

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

GRAPHIC PROCESSING METHOD AND DEVICE THEREFOR

Patent Number: JP8096025
Publication date: 1996-04-12
Inventor(s): TOKUMASU SHINJI; NONAKA SHIRO; KAWASHIMA YASUMASA
Applicant(s): HITACHI LTD
Requested Patent: ☐ JP8096025
Application Number: JP19950151438 19950619
Priority Number(s):
IPC Classification: G06F17/50 ; G06T1/00
EC Classification:
Equivalents:

Abstract

PURPOSE: To provide a graphic processing method/device which can be practically used based on the relative position data by attaining a simple magnification processing that requires no exceptional processing, display processing of a high processing speed and high accuracy, and communication control respectively.

CONSTITUTION: The object data consisting of a graphic name, the generation procedure address, the address of an operand graphic name and a parameter are generated in a graphic object generation procedure s110. In a graphic display procedure s130 where the display message of a graphic G is received from a transmission control function 11, a space point P is sent to a relative position data generation procedure s120 and the relative position data to the point P are received for the graphic G. When the graphic G is identical with an operand graphic, the point P is sent to a graphic of a lower layer so that the relative position data are obtained. In the procedure s130, a repeating point P is set in the line of sight direction for reception of the relative position data, the boundary points of the graphic G are obtained at a point P where the inside/ outside decision data I0 is inverted, and furthermore, a train of boundary points is displayed on a screen with shift of the line of sight.

Data supplied from the esp@cenet database - I2

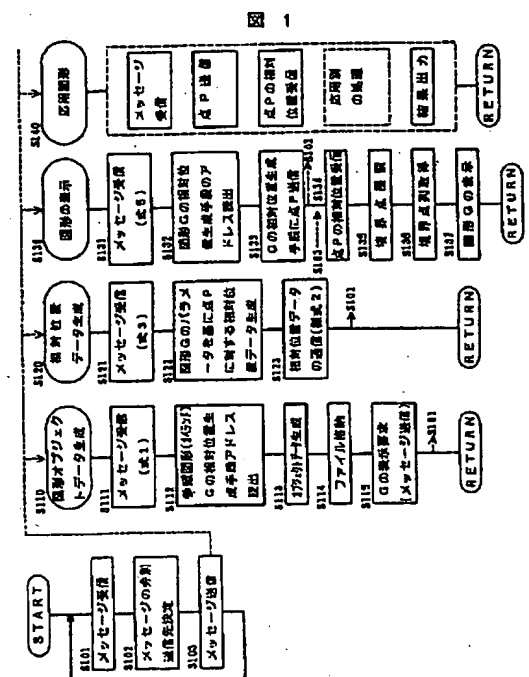
(11)特許出願公開番号

(43)公開日 平成8年(1996)4月12日

(51)Int.Cl.*	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 17/50				
G 0 6 T 1/00				
		9191-5H	G 0 6 F 15/ 60	6 2 6 Z
		9191-5H		6 2 6 C
		9365-5H	15/ 62	K
			審査請求 未請求 請求項の数34	OL (全 66 頁)

(74)代理人 弁理士 高橋 明夫 (外1名)

【構成】図形オブジェクト生成手順 110 は、図形名、生成手順アドレス、オペランド図形名のアドレス、パラメータからなるオブジェクトデータを生成する。送信制御機能 11 から、図形 G の表示のメッセージを受信した図形の表示手順 130 は、相対位置データ生成手順 120 に空間点 P を送り、図形 G に対する点 P の相対位置データを受け取る。図形 G がオペランド図形のときは、さらに下層の図形に点 P を送って相対位置データを求める。表示手順 130 は、視線方向に繰返し点 P を設定して相対位置データを受信し、その内外判定データ I O が反転する点 P から図形 G の境界点を求め、さらに視線を移動して取得した境界点列を画面に表示する。



【特許請求の範囲】

【請求項1】 図形領域が任意の空間点の相対位置データによって表され、ある図形領域の相対位置データが一つまたは一以上の他の図形領域を基にした所定の生成処理によってなる図形処理方法において、

前記所定の生成処理を行う所定図形の相対位置データ生成機能（以下、生成機能）は、前記他の図形領域の相対位置データDを生成する生成機能に前記空間点を送信し、それより前記他の図形領域に対する前記空間点Pの相対位置データDを受信し、相対位置データDは空間点Pが前記他の図形領域の内部（IN）か外部（OUT）かを示す内外判定データIOと、前記他の図形領域の境界と空間点Pとの最短距離または該最短距離以下で少なくともどの程度離れているか（距離下界）の一方を示す距離データdと、前記最短距離が設定されるときその近地点Qと、近地点Qが設定されるときを第一種、そうでないときを第二種とする種別フラグKINDを含み、この相対位置データDを基に、前記ある図形領域に対する空間点Pの相対位置データDDを生成することを特徴とする図形処理方法。

【請求項2】 請求項1において、前記所定の生成処理は前記他の図形領域の反転であり、相対位置データDの前記内外判定データの値のみを反転して前記ある図形領域の相対位置データDDを生成することを特徴とする図形処理方法。

【請求項3】 請求項1において、前記所定の生成処理は二つの他の図形領域の和（+）であり、前記他の図形領域各々の相対位置データDを基に和の場合分け処理を行って、前記ある図形の相対位置データDDを生成することを特徴とする図形処理方法。

【請求項4】 請求項1において、前記所定の生成処理は前記他の図形領域を所定距離広げる拡大処理であり、相対位置データDを基に拡大の場合分け処理を行って、前記ある図形の相対位置データDDを生成することを特徴とする図形処理方法。

【請求項5】 請求項4において、前記拡大処理は操作図形方式であり、前記所定距離をパラメータとする点対称の操作図形の中心を前記他の図形の近地点Qと一致するように対応させ、前記他の図形の境界の全体または所定の範囲に渡って前記点対象図形の集合和をとるように処理することを特徴とする図形処理方法。

【請求項6】 請求項4において、前記拡大処理は操作図形方式であり、半径を前記所定距離、中心を近地点Qとする円または球を操作図形として、相対位置データDDを生成することを特徴とする図形処理方法。

【請求項7】 請求項6において、前記操作図形方式は、前記相対位置データDがKIND＝第一種で受信されるとき、その近地点Qを中心とし前

記所定距離rを半径とする円図形と空間点Pと近地点Qを結ぶ半直線PQとの交点Rを求め、前記他の図形に対する交点Rの相対位置データD1（＝IO1, d1, Q1, 1）を求めて、以下に示す前記拡大の場合分けを行うことを特徴とする図形処理方法。

（1）前記内外判定データIO＝INの場合は、距離d1が所定距離rに等しいとき、相対位置データDDの内外判定データを内部（IN）、距離データをd+r、近地点を交点Rとする。そうでないとき及びKIND＝第二種の場合は、近地点＝nil（不定）として、相対位置データDDを第二種で表す。

（2）前記内外判定データIO＝OUTの場合は、相対位置データDDの内外判定データを外部（OUT）、距離データをd-r、近地点を交点Rとする。KIND＝第二種の場合は、近地点＝nil（不定）として、相対位置データDDを第二種で表す。

（3）前記内外判定データIO＝OUTでd<rの場合は、距離d1が所定距離rに等しいとき、相対位置データDDの内外判定データを内部（IN）、距離データをd-r、近地点を交点Rとする。そうでないときは、近地点＝nil（不定）として、相対位置データDDを第二種で表す。但し、KIND＝第二種の場合は、前記他の図形領域に対する所定の判定機能によって前記空間点Pの内外判定データIOPと距離データdPを求め、これらを含んで相対位置データDDを生成する。

【請求項8】 請求項7において、前記所定の判定機能は、空間点Pを中心とする半径rの所定図形と前記他の図形が交差する場合にIOP＝INとする超球判定機能によることを特徴とする図形処理方法。

【請求項9】 図形領域Gに対する任意の空間点Pの相対位置を表す相対位置データDを生成し、このデータを用いて図形領域Gに関わる所定の図形処理を行う図形処理方法において、

相対位置データDは、空間点Pが図形領域Gの内部（IN）または外部（OUT）を示す内外判定データIOと、空間点Pと図形領域Gの境界との最短距離または距離下界で示す距離データdと、前記最短距離が設定されるときその近地点Qと、近地点Qが設定されるときを第一種、そうでないときを第二種とする種別フラグを含み、

前記所定の図形処理は拡大処理であり、図形領域Gの境界から距離rだけ拡大する図形領域G*を生成する場合に、図形領域Gの相対位置データ生成機能（以下、図形G生成機能）に空間点Pを与えて相対位置データDを受けとり、その内外判定データIOが外部（OUT）で且つ、距離dが距離rより小さいとき、図形領域Gに対する超球判定機能によって空間点Pの内外判定データIOPと距離データdPを求め、拡大図形領域G*に対する空間点Pの相対位置データDDの内外判定データをIO

3

P、距離を d_P として生成することを特徴とする図形処理方法。

【請求項10】 請求項9において、
前記超球判定機能は、空間点Pを中心とし距離 r を半径とする円領域または球領域と図形領域Gとの交差の有無を判定し、交差する場合に内外判定IOPを内部(IN)、交差しない場合に外部(OUT)とすることを特徴とする図形処理方法。

【請求項11】 請求項10において、
前記交差の判定は、前記円領域の円周上に複数の設定点Tをモンテカルロ法を用いて順次に設定し、前記図形G生成機能による設定点Tの内外判定データIOTがINとなるとき前記交差が有ると判定し且つ、当該設定点Tの距離データ d_T を距離データ d_P となし、一方、所定数の設定点Tの内外判定データIOTが全てOUTのとき前記交差が無いと判定し且つ、設定点Tによる最短の距離データ d_T を距離データ d_P となすことを特徴とする図形処理方法。

【請求項12】 請求項10において、
前記交差の判定は、前記球領域の球面上に最大N個のランダム点Tを、1点当たり2個発生する一様乱数から定まる座標値により順次に設定し、前記図形G生成機能による設定点Tの内外判定データIOTがINとなるとき前記交差が有ると判定し且つ、当該設定点Tの距離データ d_T を距離データ d_P となし、一方、所定数の設定点Tの内外判定データIOTが全てOUTのとき前記交差が無いと判定し且つ、設定点Tによる最短の距離データ d_T を距離データ d_P となすことを特徴とする図形処理方法。

【請求項13】 請求項10において、
前記交差の判定は、予め、前記球領域の球面を北極N及び南極Sを通る経度 $\pm\pi$ の経線に基づいて複数の矩形領域に展開した球面展開図上において、球面上の設定点Tを経度 θ と緯度 ϕ による座標によって対応付けておき、前記球面展開図に四分木法を作用させて、交差する矩形領域を順次細分して絞り込みを行うことを特徴とする図形処理方法。

【請求項14】 請求項4～13のいずれか1項において、
前記拡大処理は、図形領域Gの境界から距離 r だけ拡大するオフセット処理であることを特徴とする図形処理方法。

【請求項15】 図形領域Gに対する任意の空間点Pの相対位置を表す相対位置データを生成し、このデータを用いて図形領域Gに関わる所定の図形処理を行う図形処理方法において、

前記所定の図形処理として図形領域Gを表示する場合に、図形領域Gに対する所定の視線方向(view)を設定し、前記視線方向の視線上に空間点Pを開始点から所定移動距離毎に順次設定して、図形領域Gの相対位置

4

データ生成機能に送信しては各空間点Pの相対位置データを受信し、その相対位置データに含まれる内外判定データIOが反転する前後の空間点Pから当該視線における図形領域Gの境界点を算出し、複数の視線について同様にして得られる境界点列を結んで図形領域Gを表示することを特徴とする図形処理方法。

【請求項16】 請求項15において、
前記所定移動距離は、空間点Pの相対位置データに含まれる距離データ d 以下で、所定の交差感度 ϵ 以下となるときには交差感度 ϵ とすることを特徴とする図形処理方法。

【請求項17】 請求項16において、
前記複数の視線は視線方向と垂直方向で、その走査幅Wは交差感度 ϵ 以下とされることを特徴とする図形処理方法。

【請求項18】 請求項15、16または17において、
前記視線毎の前記開始点は、前回の視線上で前記境界点が求められた場合に、当該境界点から今回の視線上へ下した垂点または前回の境界点と前回の境界点を結ぶ直線と今回の視線との交点にとることを特徴とする図形処理方法。

【請求項19】 請求項15～18のいずれか1項において、
前記境界点列の中で、結ぶ相手が不明な境界点のあるときは、前記視線方向(view)を変更して、前記境界点を算出する処理を繰返し行うことを特徴とする図形処理方法。

【請求項20】 図形領域Gに対する任意の空間点Pの相対位置を表す相対位置データを生成し、このデータを用いて図形領域Gに関わる所定の図形処理を行う図形処理方法において、

前記所定の図形処理として図形領域Gを表示する場合に、図形領域Gを囲む正方領域を四分木法により分割し、分割された四つの正方領域毎の各中心点Pを図形領域Gの相対位置データ生成手段(以下、図形G生成手段)に送信して各々の相対位置データを受けとり、該データに含まれる内外判定データIOと距離 d に基づいて、各々の正方領域が図形領域Gの内部領域か外部領域かそのいずれでもない未定領域か判定し、前記未定領域を四分木法により再分割して上記一連の処理を繰返し、分割された正方領域が所定領域感度 δ 以下になったとき、残っている未定領域毎にその4頂点の各々を前記図形G生成手段に送信し、その各相対位置データを受信して前記4頂点の内外判定データIOからIO組合せパターンを定め、予めIO組合せパターン毎に設定されている表示セグメントを対応して、前記残っている各未定領域の表示セグメントを結んで図形領域Gを表示することを特徴とする図形処理方法。

【請求項21】 請求項20において、

前記IO組合せパターンは、前記4頂点の内外判定データの全部が内部(IN)または外部(OUT)、前記4頂点の内外判定データの中の3頂点が内部(IN)または外部(OUT)、前記4頂点の内外判定データの一边の2頂点が同一、前記4頂点の内外判定データの対角点が同一によって区分されることを特徴とする図形処理方法。

【請求項22】 図形領域Gに対する任意の空間点Pの相対位置を表す相対位置データを生成し、このデータを用いて図形領域Gに関わる所定量を計算する図形処理方法において、

前記所定量として図形領域Gの面積を計算する場合には、図形領域Gを囲む正方領域を四分木法により分割し、または、前記所定量として図形領域Gの体積を計算する場合には、図形領域Gを囲む正方領域を八分木法により分割し、分割された正方領域毎の各中心点Pを図形領域Gの相対位置データ生成手段(以下、図形G生成手段)に送信して各々の相対位置データを受けとり、該データに含まれる内外判定データIOと距離dに基づいて、各々の正方領域が前記図形領域Gの内部領域または外部領域またはそのいずれでもない未定領域となるか判定し、前記未定領域となる各正方領域を四分木法または八分木法により再分割して上記一連の処理を繰返し、分割された正方領域が所定領域感度 δ 以下になったとき、上記の処理過程で前記内部領域と判定された全ての正方領域の面積または体積の総和を求めることを特徴とする図形処理方法。

【請求項23】 請求項22において、前記正方領域が所定領域感度 δ 以下になったときの未定領域は、その中心点Pの内外判定データIOがINのとき、前記内部領域とすることを特徴とする図形処理方法。

【請求項24】 所定の図形が対応される空間点の相対位置データによって表わされ、該相対位置データの点列によって前記所定図形を生成する図形処理方法において、任意の空間点Pを受信したときに、この空間点Pが前記所定図形の内部(IN)か外部(OUT)かを示す内外判定データIOと、前記所定図形の境界と空間点Pとの最短距離または該最短距離以下で少なくともどの程度離れているか(距離下界)の一方を示す距離データdと、前記最短距離が設定されるときその近地点Qと、近地点Qが設定されるときを第一種、そうでないときを第二種とする種別フラグKINDからなる相対位置データDを生成する詳細計算と、

前記所定図形に関して参照可能に設定されている領域データが使用可能な範囲においては、前記領域データを基に求めた内外判定データIO及び距離データdと、第二種の種別フラグKINDを含む相対位置データDを生成する簡略計算を、使い分けながら処理することを特徴と

する図形処理方法。

【請求項25】 請求項24において、前記領域データは、前記所定の図形を含む近似的に極小で単純な領域図形(円または四角形)でなる静的領域データであり、前記使用可能な範囲は前記空間点が前記領域図形の外部にある場合であることを特徴とする図形処理方法。

【請求項26】 請求項24または25において、前記領域データは、最初に受信した空間点P0の相対位置データDの詳細計算の結果による点P0の内外判定データと距離データd0を含む動的領域データであり、前記使用可能な範囲は受信される空間点Piの点P0からの移動距離が距離データd0以内にある場合で、これを超える空間点Pについては再度詳細計算を行ってその結果により前記動的領域データも更新することを特徴とする図形処理方法。

【請求項27】 請求項1~23のいずれか1項の図形処理方法において、

前記相対位置データの生成は、請求項24、25または26の図形処理方法によって行うことを特徴とする図形処理方法。

【請求項28】 任意の図形領域に対する任意の空間点の相対位置データを用いて前記図形領域に関わる所定の図形処理を行う図形処理装置において、

第一の図形領域の相対位置データを生成する第一の図形生成手段と、前記第一の図形領域の相対位置データを基に所定の生成処理をして第二の図形領域の相対位置データを生成する第二の図形生成手段を含む図形生成手段と、

前記第一の図形領域及び/又は前記第二の図形領域を表す相対位置データを受信してその図形領域を表示する表示処理手段と、

前記第二の図形領域に関わる前記所定の図形処理の指示に従い、所定の空間点を前記第二の図形生成手段、さらに前記第一の図形生成手段へと送信して各々を実行し、前記第一の図形生成手段から第一の相対位置データを前記第二の図形生成手段へ、前記第二の図形生成手段から第二の相対位置データを前記表示処理手段へと送信する通信制御手段と、を備えることを特徴とする図形処理装置。

【請求項29】 任意の図形領域に対する任意の空間点の相対位置データを用いて前記図形領域に関わる所定の図形処理を行う図形処理装置において、

円、四角形等の基本図形領域毎の基本図形生成手順に従って相対位置データを生成する基本図形生成手段と、反転図形生成手順に従って任意の図形領域の反転図形領域の相対位置データを生成する反転図形生成手段、和

(+) 図形生成手順に従って二つの図形領域の和図形領域の相対位置データを生成する和図形生成手段、拡大図形生成手順に従って任意の図形領域の拡大図形領域の相

対位置データを生成する拡大図形生成手段の少なくとも一つと、を含む図形生成処理手段と、
前記基本図形生成手順と、前記反転図形生成手順、前記和(+)図形生成手順、前記拡大図形生成手順の少なくとも一つと、を記憶する生成手順記憶エリアと、前記各図形領域を示す図形名、その図形生成手順のアドレス、パラメータ及び/又はオペランドの図形オブジェクトデータのアドレスを含む階層構成の図形オブジェクトデータを記憶するオブジェクトデータエリアを有する記憶装置と、
表示処理手順などの応用処理手順を有し、任意の図形領域の相対位置データを基に、前記応用処理手順に従った図形処理を行う応用図形処理手段と、
前記所定の図形処理の指示による処理内容と対象図形名の受信と、前記処理内容に対応する前記応用処理手順の実行、前記対象図形名の図形オブジェクトデータの読み出し、そのオブジェクトデータの図形生成手順の実行、そのオブジェクトデータにオペランドの図形のあるとき更にその図形生成手順の実行、前記応用処理手順から各図形生成手順に前記階層構成の順序に前記空間点を含むメッセージの送信と、前記順序と逆に各図形生成手順による相対位置データの返信と、を行う通信制御手段を設けることを特徴とする図形処理装置。

【請求項30】 請求項28または29において、
所望の図形名とオペランドの図形名を含む図形オブジェクトデータ生成のメッセージを受信し、前記所望の図形名の図形領域を生成するための生成手順のアドレスを前記生成手順記憶エリアから読み出し、前記オペランドの図形名のオブジェクトデータのアドレスを前記オブジェクトデータ記憶エリアから読み出し、前記所望の図形名、前記生成手順のアドレス及び前記オペランドの図形名のオブジェクトデータのアドレスを含むオブジェクトデータを生成し、前記オブジェクトデータ記憶エリアに記憶する図形オブジェクトデータ生成手段を備えることを特徴とする図形処理装置。

【請求項31】 請求項29において、
前記拡大図形生成手順は、任意の図形領域と前記空間点が特定の位置関係になるとき、前記拡大図形領域の相対位置データの内外判定データと距離データを求める超球判定手順を有して構成されることを特徴とする図形処理装置。

【請求項32】 請求項31において、
前記超球判定手順は、並列処理するように構成されることを特徴とする図形処理装置。

【請求項33】 請求項29において、
前記応用処理手順は、NC工作機械の工具経路データの生成処理として適用されることを特徴とする図形処理装置。

【請求項34】 請求項28または29において、
前記表示処理手順は、以下の(1)または(2)の表示

手段を選択利用可能に含んでなることを特徴とする図形処理装置。

(1) 任意の図形領域に対する所定の視線上で前記空間点を移動設定しつつ、相対位置データの内外判定データが反転する空間点より、当該視線上で図形領域の境界点を求める視線式表示手段。

(2) 任意の図形領域を囲む正方領域を四分木法により分割し、その各正方領域の中心を前記空間点とする相対位置データの内外判定データから図形領域に対する各正方領域の内、外、未定の領域判定を行い、未定の領域について上記一連の処理を所定領域感度まで繰り返した後、残っている未定の領域の4頂点について得た内外判定データの組合せパターンに応じて予め定めた所定の表示セグメントを当該未定領域の表示線とする四分木式表示手段。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、CADやCAMあるいはコンピュータグラフィックス(CG)における図形処理方式に係り、特に相対位置データを用いた図形の生成、表示、応用処理等の図形処理方式に関する。

【0002】

【従来の技術】 近年、CADやCAMあるいはCG等の図形処理技術が目ざましく進歩し、二次元や三次元での領域の定義、加工、編集、表示等に種々の方式が提案され、実用されているものも多い。

【0003】 たとえば、三次元の図形処理に関しては、バウンダリー・リプレゼンテーション法、コンストラクティブ・ソリッド・ジオメトリ法、オクトリー法が代表的で、これらに関する多数の文献が出版されている。

【0004】 応用図形処理についても同様である。たとえば、与えられた図形を外側または内部側に一定の距離だけ拡大するオフセット技術一つみても、数値制御(NC)の軌跡データ作成などに関して、特開昭58-181185号公報、特開昭60-108222号公報、特開昭60-184484号公報など、多数の出願がみられる。また、オフセットを利用して、製品形状に現われる凸部や凹部の角に丸み付けるフィレット処理に関して、特開平3-228181号公報等があり、さらに別の応用として、オフセットされた図形間で交差チェックを行い物体間の干渉を判定する特公昭61-533号公報等と枚挙にいとまがない。

【0005】 しかし、これらの従来の図形処理技術は、予め対象とする図形の境界が陽に分かっていることが前提である。ここで、陽に分かるとは、図形の境界が位置や形の情報(境界上の各点に於ける傾きや曲率などの微分幾何学的情報)によって表されることを意味する。

【0006】 たしかに、円や四角形のように境界に関するパラメータや、方程式が分かっている場合は、それをそのまま図形処理に用いるのが効率的である。しかし、

図形がこれら単純な図形の複雑な組合せでなり、さらにそのオフセット図形であったりすると、境界を表す情報が得られなかったり、たいへんに複雑な式の組合せとなつて処理が困難になり、図形処理方法の抱える大きな問題点となっている。

【0007】これに対し、複雑な図形の境界の位置や形の情報が陽に分からない場合の図形処理方法として、先に本発明者等が提案した相対位置データを利用する新規な方法がある（特開平 2-7174号公報；以下では引用例と呼ぶ）。

【0008】この方法は、対象図形Gに対する任意の空間点Pの相対位置データを利用するもので、それには図形Gに対する空間点Pの内／外を判定するデータが含まれている。生成される図形オブジェクトデータは、円や三角等のように本来、境界が陽に分かる基本図形を最下層とし、その上に組合せや反転あるいは拡大等により生成される図形を積み重ねて、階層構成に作成される。

【0009】相対位置データを用いた図形処理として、例えば和図形Gを表示する場合、図形Gからそのオペランド図形（和図形を構成している二つの図形）に空間点Pを送って各図形から相対値データを受け取り、和図形生成手順に従って一方の相対位置データを採用する。そのデータに従い、P点を中心とする一定領域が図形Gの内部／外部かを判定し、どちらとも決定できない場合、その未定領域を四分木法で次々と細分していき、最終的に内部／外部に分かれるところを境界線として表示する。

【0010】

【発明が解決しようとする課題】上記の引用例に開示している相対位置データによる図形処理方法によれば、複雑な図形の境界の位置や形の情報が陽に分からなくても、その図形を構成する最下層の図形（ツリーの葉の部位）部分からの相対位置データを基にして、表示をはじめ種々の図形処理が可能になる。しかし、上記引用例の方法は実用化に際していくつかの問題点が明らかになっている。

【0011】その第一は、上層の図形オブジェクトデータが下層のオペランドを含む場合等に、各層の図形間での空間点やパラメータあるいは相対位置データなどを動的に接続する制御方法が不明で、複数の階層に跨るダイナミックな図形処理の方法と装置の構成が提供されていないことである。

【0012】第二には、拡大処理（オフセットを含む）において、第二種の相対位置データに距離dの上界の概念を含むために、場合分けの処理が複雑化し不正確となる場合があることである。特に、元図形の境界に近い外部に空間点Pが位置する場合には、この点Pの内外判定には結果的に、点Pと図形境界との真の距離を求める必要があることである。この真の距離の算定は最小化問題のため、特別な場合を除いて一般には難しく、複雑なス

テップを必要とする。結局、相対位置データを用いる拡大の図形処理に処理不可能な場合を生じ、図形処理の信頼性の低下ないしは適用範囲が制約される問題点がある。

【0013】第三に、画像処理は一般にオペレータとの対話によることが多く、その処理過程で常に表示処理が伴う。しかし、上記した従来の四分木法による表示は、境界が陽に分かるものに比べ処理時間がかかり、境界線の表示精度にも問題があり、実用上の大きなネックとなっている。

【0014】第四に、相対位置データ生成による画像処理の速度が実用の域に達していないことにある。

【0015】本発明の第一の目的は、従来技術の問題点を克服して階層関係にある図形間の動的な通信制御方式を与え、相対位置データを用いる実用可能な図形処理方法および装置を提供することにある。

【0016】本発明の第二の目的は、相対位置データを用いる拡大を含むあらゆる処理を効率的に且つロバストに実現でき、信頼性が高く使い勝手のよい図形処理方法および装置を提供することにある。

【0017】本発明の第三の目的は、相対位置データを用いる図形処理において、表示処理の高速化と表示精度の向上を可能にする図形処理方法および装置を提供することにある。

【0018】本発明の第四の目的は、相対位置データの生成処理を高速化して、三次元モデルを含む複雑な画像処理を実用可能な水準に引上げることにある。

【0019】その余の目的は、以下の記載を通して明らかになる。

【0020】

【課題を解決するための手段】本発明の第一の目的は、図形領域に対する任意の空間点の相対位置データを用いて前記図形領域に関わる所定の図形処理を行う図形処理装置において、第一の図形領域の相対位置データを生成する第一の図形生成手段と、前記第一の図形領域の相対位置データを基に所定の生成処理をして第二の図形領域の相対位置データを生成する第二の図形生成手段を含む図形生成手段と、前記第一の図形領域及び／又は前記第二の図形領域を表す相対位置データを受信してその図形領域を表示する表示処理手段と、前記第二の図形領域に関わる前記所定の図形処理の指示に従い、所定の空間点を前記第二の図形生成手段、前記第一の図形生成手段へと送信して各々を実行し、前記第一の図形生成手段から第一の相対位置データを前記第二の図形生成手段へ、前記第二の図形生成手段から第二の相対位置データを前記表示処理手段へと送信する通信制御手段と、を備えることにより達成される。

【0021】本発明の第二の目的は、図形領域Gに対する任意の空間点Pの相対位置を表す相対位置データDPを生成し、このデータを用いて図形領域Gに関わる所定

の図形処理を行う図形処理方法において、相対位置データDPは、空間点Pが図形領域Gの内部(IN)または外部(OUT)を示す内外判定データIOと、空間点Pと図形領域Gの境界との最短距離またはそれ以下の少なくともどの程度離れているかを示すデータ(以下、距離下界と呼ぶ)の距離データdと、前記最短距離が設定されるときその近地点Qと、近地点Qが設定されるときを第一種、そうでないときを第二種とする種別フラグを含み、前記所定の図形処理として、図形領域Gの境界から距離rだけ拡大する図形領域G*を生成する場合に、距離rをパラメータとする点対称の操作図形の中心を相対位置データDPの近地点Qと一致するように対応させ、図形領域Gの全体または所定の範囲に渡って前記対象図形の集合和をとるように処理することにより達成される。

【0022】上記における拡大の場合分けは、前記操作図形を円図形とする場合で、相対位置データDPがKIND=第一種で受信されるとき、その近地点Qを中心とし距離rを半径とする円図形と空間点Pと近地点Qを結ぶ半直線PQとの交点Rを求め、図形領域Gに対する交点Rの相対位置データD1(=IO1, d1, Q1, 1)を求めて、以下の(1)~(3)の拡大の場合分けによって拡大図形領域G*の相対位置データDDを生成することを特徴とする。

【0023】(1)前記内外判定データIO=INの場合は、距離d1が距離rに等しいとき、相対位置データDDの内外判定データを内部(IN)、距離データをd+r、近地点を交点Rとする。そうでないとき及びKIND=第二種の場合は、近地点=nil(不定)として、相対位置データDDを第二種で表す。

【0024】(2)前記内外判定データIO=OUTの場合は、相対位置データDDの内外判定データを外部(OUT)、距離データをd-r、近地点を交点Rとする。KIND=第二種の場合は、近地点=nil(不定)として、相対位置データDDを第二種で表す。

【0025】(3)前記内外判定データIO=OUTでd<rの場合は、距離d1が所定距離rに等しいとき、相対位置データDDの内外判定データを内部(IN)、距離データをd-r、近地点を交点Rとする。そうでないときは、近地点=nil(不定)として、相対位置データDDを第二種で表す。但し、KIND=第二種の場合は、前記他の図形領域に対する所定の判定機能によって前記空間点Pの内外判定データIOPと距離データdPを求め、相対位置データDDの内外判定データをIOP、距離データをdPとする。

【0026】前記所定の判定機能は、空間点Pを中心とする半径rの所定図形と前記他の図形が交差する場合にIOP=INとする超球判定機能によることを特徴とする。

【0027】本発明の第三の目的は、図形処理として図

形領域Gを表示する場合に、該領域Gに対する任意の視線方向(view)を設定し、前記視線方向の視線上の空間点Pを開始点から所定移動距離毎に順次、図形領域Gの相対位置データ生成機能に送信し、前記相対位置データ生成機能から各空間点Pの相対位置データを受信し、その相対位置データに含まれる内外判定データIOが反転する前後の二つの空間点Pから当該視線上における図形Gの境界点を算出し、複数の視線について同様にして得られる境界点列を結んで前記図形Gを表示することにより達成される。

【0028】あるいは、前記図形領域Gを表示する場合に、該領域Gを囲む正方領域を四分木法により分割し、分割された四つの正方領域毎の各中心点Pを図形領域Gの相対位置データ生成手段(以下、図形G生成手段)に送信して各々の相対位置データを受けとり、該データに含まれる内外判定データIOと距離dに基づいて、各々の正方領域が図形領域Gの内部領域または外部領域またはそのいずれでもない未定領域となるか判定し、判定された未定領域を四分木法により再分割して上記一連の処理を繰返し、分割された正方領域が所定領域感度δ以下になったとき、残っている未定領域毎にその4頂点の各々を前記図形G生成手段に送信し、その各相対位置データを受信して前記4頂点の内外判定データIOからIO組合せパターンを定め、予めIO組合せパターン毎に設定されている表示セグメントを対応して、前記残っている各未定領域の表示セグメントを結んで図形領域Gを表示することにより達成される。

【0029】本発明の第四の目的は、所定の図形が対応される空間点の相対位置データによって表わされ、該相対位置データの点列によって前記所定図形を生成する図形処理方法において、任意の空間点Pを受信したときに、この空間点Pが前記所定図形の内部(IN)か外部(OUT)かを示す内外判定データIOと、前記所定図形の境界と空間点Pとの最短距離または該最短距離以下で少なくともどの程度離れているか(距離下界)の一方を示す距離データdと、前記最短距離が設定されるときその近地点Qと、近地点Qが設定されるときを第一種、そうでないときを第二種とする種別フラグKINDからなる相対位置データDを生成する詳細計算と、前記所定図形に関して参照可能に設定されている領域データが使用可能な範囲においては、前記領域データを基に求めた内外判定データIO及び距離データdと、第二種の種別フラグKINDを含む相対位置データDを生成する簡略計算を、使い分けながら処理することにより達成される。

【0030】前記領域データは、前記所定の図形を含む近似的に極小で単純な領域図形(円または四角形)となる静的領域データであり、前記使用可能な範囲は前記空間点が前記領域図形の外部にある場合であることを特徴とする。

13

【0031】あるいは、前記領域データは、最初に受信した空間点P0の相対位置データDの詳細計算の結果による点P0の内外判定データと距離データd0を含む動的領域データであり、前記使用可能な範囲は受信される空間点Piの点P0からの移動距離が距離データd0以内にある場合で、これを越える空間点Pについては再度詳細計算を行ってその結果により前記動的領域データも更新することを特徴とする。

【0032】

【作用】本発明で用いられる相対位置データは、図形の頂点等の座標でも境界線の方程式でもなく、対象の図形Gに対して任意の空間点Pが内部(IN)にあるか外部(OUT)にあるかを示す内外判定データ(IO)、図形Gの境界と点Pの最短距離またはそれ以下で境界から少なくともどの程度離れているかを示す距離データ

(d)、距離dが最短距離の場合の近地点Qなどを要素とする。

【0033】図形の位置や形の情報を陽に含まない相対位置データを用いることは、レーダーで電波を発信してその進行方向にある物体の境界点をみつけ、その境界点を結んで物体の形状を認識する場合に例えることができる。

【0034】このように、存在している図形領域の境界は分かるが数式などによる表現が困難な図形領域に対し、相対位置データを用いてその形状や境界を生成すること、言い替えれば図形生成のための相対位置データを生成する図形処理方式が、本発明の基本的なメカニズムとなっている。その場合、空間点と対象図形領域と所定距離以上離れている場合は、領域データによる簡略計算が適用され、高速化が図られる。

【0035】本発明の図形処理装置によれば、図形の生成手段の相互間あるいはそれら生成手段と表示処理などの応用処理手段の間で、上層から下層へ空間点Pなどがメッセージされ、反対に下層から上層へと相対位置データが求められて応用処理手段へとメッセージされるダイナミックな通信制御が可能になるので、階層関係に有る図形の相対位置データによる生成と応用図形処理が効率的に実現できる。

【0036】本発明の拡大図形(G*)の生成によれば、上記した操作図形方式による場合分けを行うので、引用例の場合に比べ単純化されるうえに、距離d≠拡大距離rの場合にも正確な相対位置データを求めることができる。

【0037】さらに、空間点Pが元の図形Gの外部でその境界の近く(拡大距離rの内側)にあるときも、上記の超球判定法によって点Pの内外判定が行えるので、どのような図形の場合にも例外なく拡大処理が可能になり、相対位置データを用いる図形処理の信頼性を高め、適用範囲の制約の払拭を可能にする。さらに、本発明の図形処理で時間のかかる超球判定等の繰返し処理部は、

14

並列処理化によって実用可能な処理速度を実現している。

【0038】本発明の表示処理によれば、視線方式による境界探索によって相対位置データを用いる表示速度を実用可能なレベルに向上できる。また、再帰的な四分木方式による未定領域の組合せパターンから選択される表示セグメントをもちいるので、従来の四分木方式に比べ処理が速く且つ、図形精度の高い表示が実現できる。これら表示法を選択可能に用いることで、表示処理を不可欠とする大多数の図形処理における相対位置データの適用を可能にする。

【0039】本発明は特に三次元モデル、たとえばNC工作機械の工具経路の計算、凹凸部のフィレット処理、移動体の緩衝チェックなどの画像処理において好適である。

【0040】

【実施例】以下、本発明の実施例を図面を用いて詳細に説明する。

【0041】はじめに、実施例の図形処理装置の構成と動作の概略を説明する。図2は、図形処理装置の機能ブロック図で、中央処理装置(CPU)1、記憶装置2、入力装置3及び表示装置4からなる計算機装置によって構成される。

【0042】CPU1は、入力装置や各処理機能とメッセージを送受しながら、各処理機能を制御する通信制御機能11、図形オブジェクトデータを作成する図形オブジェクト作成機能12、図形の相対位置データを生成する図形相対位置生成機能13、図形の表示を処理する図形表示機能14、所定の応用図形処理を行う応用図形処理機能15を有している。

【0043】応用図形処理機能15は、図形オブジェクトデータと図形相対位置生成機能を利用して行う図形処理で、たとえば面積計算、NC情報の生成、配置や干渉チェックなど多種多様である。表示も応用処理の一つであるが、本実施例では説明の都合上から独立して扱っている。

【0044】図形相対位置生成機能13は、図形に応じた相対位置データを指示点に対応して出力する処理で、生成した相対位置データは表示処理や応用処理によって具現化される。従って、図形相対位置生成機能は、本質的には相対位置データによる図形生成の機能であるが、本実施例では図形オブジェクトデータ生成との混同をさけるために、このように呼ぶことにする。

【0045】本実施例の図形相対位置生成機能13は、以下の(1)～(4)に大別されるが、(1)の基本図形に対する機能はさらに円、四角、三角等の形状別に分かれる。(2)～(4)の機能は、いわゆるオペランド図形に対する機能で、これより基本的にはあらゆる図形の生成が可能になるが、別に差図形、積図形などの処理機能を有してもよい。

15

【0046】(1) 基本図形Gの相対位置を生成する機能。

【0047】(2) 図形Gの反転図形G'の相対位置を生成する機能。

【0048】(3) 二つの図形G1, G2を集合した和図形G3の相対位置を生成する機能。

【0049】(4) 図形Gの拡大図形G*の相対位置を生成する機能。

【0050】記憶装置2には、通信制御のプログラムを記憶する通信制御手順ファイル21、図形オブジェクトデータを生成する制御プログラムを記憶する図形オブジェクト生成手順ファイル22、図形毎の相対位置データ生成の制御プログラムを記憶する相対位置生成制御手順ファイル23、図形オブジェクトデータを記憶する図形オブジェクトファイル24、図形表示の制御プログラムを記憶する表示制御手順ファイル25、用途毎に応用図形処理の制御プログラムを記憶する応用図形制御手順ファイル26を有している。

【0051】図3(a)は、図形相対位置生成手順ファイル23で、円の生成手順231、四角形の生成手順232、三角形の生成手順234等の基本図形と、反転図形の生成手順235、和(+)図形の生成手順236、拡大図形の生成手順237等(基本図形に対し、以下ではオペランド図形の生成手順と呼ぶ)が記憶されている。さらに、拡大図形の生成過程で、特殊な場合に相対位置データの内外判定に使用される超球判定処理手順238も格納している。

【0052】図示の「#×××」は、格納先頭番地であり、CPU1によって管理される。実際の格納では、手順名称と先頭アドレスの対照をヘッダー部とし、手順はその下層にアドレス毎に記憶される。

【0053】図3(b)は、図形オブジェクトデータファイル24で、図形名、その相対位置生成手順の格納番地を示す手順ポインタ、オペランド(参照する図形オブジェクトのポインタ)及びパラメータエリアからなる。*

[手順名称; パラメータ; 図形オブジェクト名称(図形名)] ... (様式1)

手順名称は、図形オブジェクト生成の場合、基本図形では「円」や「四角」など、オペランド図形では「反転」や「和(+)」などが記述される。一方、表示処理の場合は「表示」、応用処理の場合は「面積」などと記述される。オペランドで与えられる図形オブジェクト名称(以下、図形名)は、その手順の操作対象となる図形名であり、基本図形オブジェクトを生成する場合にはない。

【0061】図形オブジェクトデータ生成処理(s110)が起動され、(様式1)のメッセージが受信される(s111)。オペランドに図形名Gがある場合は、図※

[生成した図形名; 相対位置生成手順ポインタ; パラメータ;

図形ポインタ] ... (様式2)

上記和図形の例で示すと、(C; 手順#ccc, 手順# 50 sss)となり、パラメータは必要ない。和図形Cの重

16

*基本図形はオペランドを持たない。例えば図形名が「円」の場合、手順ポインタは「#ccc」を、パラメータは中心座標値と半径の値を記憶する。

【0054】図4(a)は、表示制御手順ファイル25で、図形表示処理の高速化を可能にする視線式表示処理手順251と、その探索点決定処理手順252、更には如何なる形状の表示にも適用でき且つ効率的な四分木パターン形処理手順253が、各アドレス領域に格納されている。

10 【0055】視線式表示処理手順251と四分木パターン形処理手順253は、図形表示機能14がコマンドの指示に従って選択する。もちろん、図形オブジェクトデータの生成時に、特定の図形にたいしては属性として表示手順名を持たせるようにしてもよい。

【0056】同図(b)は、応用図形制御手順ファイル26で、面積計算の手順261やNC情報の生成手順262などが各アドレス領域に格納され、更にアプリケーションに応じて種々の手順が追加される。

【0057】このように構成される図形処理装置は、オペレータが入力装置3から指示したコマンドに従い、新たな図形オブジェクトの生成又は図形の表示、あるいは応用図形処理を実行する。図形の表示や応用処理では適宜、相対位置データの生成処理が行われる。各段階の処理結果は、必要に応じて表示装置4に表示される。

【0058】図1は、本実施例による図形処理装置の基本機能を示すフロー図である。通信制御機能11は、入力装置3や各処理手段からのメッセージを受信すると(s101)、メッセージの様式から通信先を決定し(s102)、通信先を呼び出して(あるいは起動して)メッセージを送信する(s103)。このとき、必要に応じてメッセージの加工が行われる。

【0059】処理態様を示すコマンド(入力装置3から)などのメッセージは、(様式1)で表される。

【0060】

※図Gのオブジェクトデータを読み出し、図形G相対位置生成手順の先頭アドレス(ファイル23)を読み出す(s112)。例えば、手順名称が和(+)で、オペランド図形名が円A、四角形Bとすると、円相対位置生成手順のポインタ#ccc、四角形相対位置生成手順のポインタ#sssが読出される。

【0062】次に、手順名称の図形オブジェクトデータを様式(2)により生成し(s113)、図形オブジェクトファイル24へ格納する(s114)。

【0063】

ね位置のパラメータは、後述する和図形の相対位置データの生成や表示の際に、基本図形A、Bの中心位置が指定されるまで、省略できる。これとは反対に、図3

(b)のように、図形によってはパラメータのみで、操作対象図形名のポインタ(先頭アドレス)が記述されないこともある。

【0064】通常、生成した図形オブジェクトデータは、通信制御機能11に表示要求のメッセージを送信して(s115)、図形の表示処理s130によって画面表示される。

【0065】このように、本実施例により生成される図形オブジェクトのデータは、図形名とその図形生成手順のポインタ(先頭アドレス)及びパラメータからなり、従来のデータのように境界や形状を陽に示すための数式などを一切持たない。

【0066】また、種々の基本図形のオブジェクトデータを下層にして、和や反転あるいは拡大などの処理を加えるオペランド図形のオブジェクトデータを、上層に積み重ねる階層構成(基本図形を葉とするツリー構造)としている。本実施例のオブジェクトデータは、このように積み重ねて生成できるので、所望の複雑な図形の生成が可能である。

【0067】一方、(様式1)の手順名称が「表示」のメッセージの場合は、図形表示処理(s120)が起動され、表示する図形名Gとパラメータが受信され(s131)、図形Gのオブジェクトデータからその相対位置生成手段のアドレス(手順名でもよい)が読出される(s132)。

【0068】次に、図形Gの相対位置生成手段に、通信制御機能11を介して(s101~s103)、パラメータと共に空間点P(境界探索開始点)を送る(s133)。そして、点Pに対する図形Gの相対位置データ(様式3)を受信し(s134)、後述する境界点探索を行う(s135)。一つの境界探索開始点に対応する境界点が求まったら上記s133~s135を繰返し、図形Gの前境界点列を取得する(s136)。最後に境界点列を結んで図形Gを画面表示する(s137)。な*

[空間点P; パラメータ(中心O, 半径r)] ... (様式3)

このメッセージより空間点Pと、円生成のパラメータ中心O、半径rを得て、点Pと円の周境界との符号付き距離dを計算する(s202)。距離dの符号を判定し(s203)、符号が正ならば、点Pの内外判定データIO="OUT"と設定する(s2031)。空間点P52がこのケースである。距離dが負またはゼロの場合には、IO="IN"を設定し、距離dの符号を反転する(s2032)。空間点P51がこのケースである。

【0077】さらに、点Pと円の周境界との近地点Qをベクトル計算により求める(s204)。最後に、種別フラグ="1"とする相対位置データを完成し、このデ

[内外判定データ(IO); 距離データ(d);

*お、本実施例の相対位置データを用いる表示処理の詳細は後述する。

【0069】図形相対位置生成処理s120は、(様式4)のメッセージにより起動され、受信したパラメータを基に、点Pに対する相対位置データを後述するように求め(s122)、通信制御機能11に返信する(s123)。

【0070】また、応用図形処理(s120)が起動されると、表示処理のs131~s134と同様の処理後に、相対位置データを利用した個別の応用処理、たとえば面積計算が行われて、結果が出力される。応用処理の詳細は後述する。

【0071】以下、具体的な図形例によって、本実施例の図形処理装置により行なわれる図形処理、即ち、図形領域の相対位置データ生成処理(図形領域生成処理)とその表示処理、さらに応用図形処理のいくつかを詳細に説明することにする。なお、以下では特に断らないかぎり空間の次元を二次元とし、図形領域は単に図形と呼んで説明する。

【0072】はじめに、相対位置データを用いる図形処理の理解を容易にするために、基本図形を例にその相対位置データの生成と表示処理について説明する。基本図形としては、四角形、三角形および円等が対象になる。

【0073】図5に、円の例を示す。円の図形オブジェクトデータは上記(様式2)で与えられ、[円; パラメータ(中心O、半径r)]となる。

【0074】図5(a)は説明図で、円50、円内部の空間点P51、外部の空間点P52を示す。また、円の中心O、半径r、近地点Q(点Pから最短の図形境界点)、線分PQの距離d(Qが分かる場合は点Pと円50の最短距離)を示す。同図(b)に円の生成手順200を示す。

【0075】円生成の制御手順s200(ファイル23の手順231)は、たとえば表示機能13から通信・制御機能11を介して起動され、(様式3)のメッセージを受信する(s201)。

【0076】

データ[IO、d、Q、1]を、通信・制御機能11を介して要求元(たとえば表示機能13)に返信する(s205)。以上の処理を、与えられる一連の空間点Pに対し繰返し行なうことで、相対位置データによる円図形が生成される。

【0078】ここで、相対位置データを定義しておく。相対位置データとは、対象とする図形Gに対して、任意の空間点Pが持つ相対的な位置関係であり、データ形式は(様式4)で表される。

【0079】

内外判定データ (IO) : 空間点Pが図形Gの中にある場合「IN」、空間点Pが図形Gの外にある場合「OUT」。

【0080】距離データ (d) : 空間点Pと図形Gの境界δGとの最短距離またはそれ以下の近似値 (距離下界)。

【0081】近地点データ (Q) : 空間点Pが境界δG上で最短距離をとる点の一つ、上記点を設定できない場合は「NIL」。

【0082】種別フラグ (KIND) : 距離データと境界点データが設定される場合「1」、それ以外の場合「2」。

【0083】なお、相対位置データは種別フラグが「1」の場合を第一種、「2」の場合を第二種と呼ぶ。また、全ての空間点Pに対して第一種の相対位置データを与える図形Gを第一種図形といい、それ以外を第二種図形と呼ぶ。第二種図形では、近地点Qは不明 (nil) のままで、距離dは最短距離より下側で、できれば近似値が表示処理等にはベターである。

【0084】計算機を用いて図形処理を行う場合、図形の境界に係るデータあるいは領域に係るデータ (マスプロパティデータ ; 図形の重心、直径、面積、体積、モーメント等) が必要になる。前者は輪郭の表示やNC制御情報の取得、後者は面積や体積の計算あるいは物体の配置や干渉解析のために用いられる。

【0085】上記した相対位置データは、メッセージによって与えられた空間点Pの一点毎に定まり、近地点Qを除く他の項目が設定される。この場合、近地点Qは分からなくても、後述するようにその内外判定から、図形の境界点やマスプロパティデータが決定できる。

【0086】ところで、円図形は明らかに第一種図形である。即ち、空間点Pと中心Oを結ぶ線分OPと半径rの差が距離dであり、線分OPまたはそのP側の延長と円周との交点が近地点Qである。内外判定データIOは、半径rと距離dの比較によって決まる。

【0087】四角形 (長方形) の場合は、その四辺を延長したときの九つの空間領域に対し、空間点Pがどの領域に入るかを判定することによって、第一種の相対位置データが得られる。他の基本図形に対しても、同様にして図形と空間点の幾何学的関係から、内外判定データIO、距離d及び近地点Qを求め得ることは自明であり、このように基本図形は、第一種の相対位置データとして求めることができる。

【0088】尤も、基本図形の表示等を行う場合は、その境界データは陽に与えられるので、従来处理の方が効率的であり、本実施例はかかる場合に従来处理の併用を排除しない。

【0089】しかし、複数の基本図形を合成した複雑な形状の図形では、その境界データが陽に分からなかった

り、分かっても求めるのに時間がかかる。このような複雑な図形は、相対位置データを用いる図形処理の好適な対象であり、第二種の相対位置データを含んで生成される場合が殆どである。第二種の相対位置データは、内外判定データのみが明確に定まり、近地点Qや最短距離は不明であるが、それでも本方式の優れた特徴によって図形処理が可能になる。

【0090】図6及び図7は、円の表示処理を説明する説明図とその相対位置データを示し、前者は第一種、後者は第二種の例である。

【0091】表示処理では、設定された視線方向 (view) に視線V1が設定され、視線V1上の開始点P0から円Aの境界位置の探索を開始する。図6の開始点P0は、図5に示した空間点P51又は52に相当する。

【0092】開始点P0の相対位置データは、図6(b)に示すように円の生成手段 (s200) から (様式4) で得られ、[OUT, c0, Q0, 1] となる。次の探索点Pは、視線V1の方向に、点P0から距離c0にある点P1に設定される。この点P1の相対位置データは、同様にして [OUT, c1, Q1, 1] となる。

点P1の距離データc1が、予め設定されている交差感度εより小さいとき、次の探索点P2は点P1から視線V1上に距離εの位置とされる。次に、探索点P2の内外判定データIOがINになると、この間に視線V1と図形Aの境界δAの交点が存在することになり、点P1と点P2に二分法を適用して境界点δ1を求める。

【0093】さらに、視線Vi、Vi+1、...と、視線方向と垂直方向に走査間隔Wづつ移動して、視線方向から見える円Aの半円部 (点線Lより上) の境界点データ列QQを求める。残る半円部の境界点は、上記で、内外判定データIOが変化した後更に視線V上を交差感度εづつ探索を続け、再びIOが変化する点を検出するようにしてもよい。しかし、一般には視線Vを再設定し、上記例では視線を下から上に向けて繰り返すことで、より効率的に残る半円部の境界点データ列を求めることができる。なお、間隔Wを交差感度εと同等または以下に設定し、表示図形が違和感なく見えるようにする。交差感度εについては後述する。

【0094】図7は、円図形Aを取って第二種図形で定義したときの表示例である。同図(b)に示すように、第二種の相対位置データにおいては、空間点Pと境界との最短距離が分からないので近地点Q=nilとなる。また、相対位置データの距離データは、P0に対してはd=c0/2が設定されている。

【0095】第一種の場合と同様に、点P0から探索を開始し、視線V1上をc0/2進んだ点P1で、[OUT, c1/2, nil, 2] が得られる。距離データc1/2が交差感度εより小さいと、次の探索点P2は点P1から視線V1上に距離εの位置となり、[OUT,

21

c2/2, nil, 2] が得られる。更に、距離 ϵ 前進した探索点P3で、その内外判定データIOがINになると、点P2とP3の二分法により境界点 $\delta 1$ が求まる。

【0096】このように、近地点Qがnil、距離dが近似値となる第二種の相対位置データによっても、基本図形の境界点を求めるのには何の不都合もない。しかし、複雑な図形の種々な場合分け処理を進めるときに、全てを第二種の相対位置データで処理しようとする処理効率が悪くなる。

【0097】本実施例では、基本図形は原則として第一*

[表示；視線方向などのパラメータ；図形名G] … (様式5)

パラメータとしては、視線方向view、開始点P0、走査線間隔幅W、走査線数N、最大探索深さL、交差感度 ϵ などである。但し、W、 ϵ などは予め定数が設定されることも多い。走査方向は、視線方向viewに垂直な長さWのベクトルで定める(s302)。

【0101】次に、探索点PをP=PP (最初は開始点P0) とし(s303)、全走査i=0~(N-1)のi=0から探索を開始する(s304)。まず、図形Gの相対位置生成手順(円の例ではS200)に探索点Pをメッセージして(s3041)、返信された相対位置データ[IOO、d、Q、KIND]を受けとる(s3042)。そのきの内外判定データIOOを判定し(s3043)、開始点P0が図形Gに対して、外にあれば探索向きVをview方向に設定し、内にあればその反対方向(V=-view)に設定して探索する。

【0102】次に、内外判定データIOOがIONに変化(IOO≠ION)するまで、または探索深さj>Lになるまで(s3044、5)、探索ステップ $\sigma = \max(d, \epsilon)$ による探索点P(=P× σ ×V)を求めながら(s30451)、図形Gの相対位置生成手順にメッセージ(探索点P)を送って(s30452)、相対位置データを受け取る(s80453)。

【0103】相対位置データを受信する度にその内外判定データを判定し(s3046)、IOOからIONに変化していれば、一つ前の探索点P0と今回の探索点Pから二分法より境界点 δi を求める(s30461)。そして、次の走査探索開始点Pを決定し(s30462)、求められた境界点 δi は境界点列QQ(i)に追加する(s3047)。一方、探索深さj>Lとなるまで内外判定データがIOOのままであれば、境界点 $\delta i = \text{nil}$ として次の探索開始点Pを決定する(s30463)。最後に、境界点列QQ(i)のi=0~(N-1)を結ぶ曲線を表示する(s305)。

【0104】本実施例の視線式表示処理方法によれば、上記引用例の四分木式表示に比べ、相対位置データを用いる場合の表示処理を大幅にスピードアップでき、十分に実用に供し得る。なお、本発明で改良した四分木式表示方法については後述する。

22

*種で表現し、複雑な図形の場合も下層の基本図形からの相対位置データを基にして、処理効率を上げ且つ、処理不能となるケースをなくしている。

【0098】図8に、本実施例の視線式図形表示の手順を示す。この手順は相対位置データを用いる場合の汎用的な方法で、基本図形には限定されない。

【0099】表示の手順s300は、表示制御手順ファイル25の処理手順251であり、(様式5)のメッセージ(基本形は(様式1)に同じ)を受信して(s301)処理を開始する。

【0100】

【0105】上記の処理(s30462、3)で、次の探索開始点Pは探索開始点P0と走査線間隔幅Wから、P=PP (最初の開始点P0) + (i+1)Wにより求めている。しかし、境界点 δi が求まったときの次の開始点は(s80462)、図形のコヒーレンス性を利用して、より効率よく決定することができる。

【0106】図9を用いて、その効率的な探索点決定方法(手順252)を概念的に説明する。今、ベクトルwを図示の矢線方向とし、視線Vi上の境界点 δi が点91に求まったとする。このとき、次の視線Vi+1の開始点P0は、点91から視線Vi+1に下ろした垂点92とすることができる。あるいは、一つ前の視線Vi-1の境界点90を起点とし、境界点91を通る半直線と次の視線Vi+1との交点93を開始点P0とすることができる。

【0107】いずれの場合にも、間隔Wを交差感度 ϵ と同等またはそれ以下の微小幅すると、次の探索点P1で内外判定データIOが反転する。従って、境界点の探索処理が大幅に効率化され、本実施例の視線式表示処理方法を、直接の境界データを用いる従来法に比べ、遜色ない程度にまで高速化できる。

【0108】図10に、交差感度の設定方法を説明する。交差感度 ϵ は、視線Vと図形Gの境界 δG の交点の検出精度を規定する値である。即ち、交差感度 ϵ とは、視線V上で任意の ϵ 幅の区間に於ては、視線と図形との交点は、高々一個であると規定する。この際、二個以上の交点がある場合でも、一個以下に縮約して扱う。同図で、区間10d、10eでは、この規定どおりになっている。しかし、区間10fでは、この規定に合致しないので、交点は一個以下に縮約される。従って、 ϵ の値を小さくする、または大きくすることは、それぞれ、交差感度を上げる、または、下げると言う意味になる。

【0109】交差感度 ϵ は通常、図形の最大寸法の1/100~1/1000程度に設定されるが、内外判定データIOが反転する2点間の距離に応じて動的に可変するようにしてもよい。

【0110】次に、オペランド図形の例として、反転図形、和(+)図形及び拡大図形の相対位置データ生成と

その表示処理について説明する。

【0111】反転図形生成の手順は、オペランドとして元図形を記述した(様式6)のメッセージを受信して実行される。

【0112】

〔反転; ; 元図形G〕 … (様式6)

反転図形生成の手順は簡単である。まず、元図形Gの相対位置データ生成手順に対して、空間点Pと元図形Gに応じたパラメータ(元図形Gが円であれば中心と半径)

からなるメッセージを送り、返信された相対位置データ〔IO, d, Q, KIND〕を受けて、その中の内外判定データIOを反転するだけでよい。

〔+; ; 図形G1、図形G2〕

和図形生成手順s400は、空間点Pと図形名G1、G2のメッセージを受信すると(s401)、図形G1と図形G2のオブジェクトデータから読出した各々の相対位置生成手順に対し空間点Pを送信する(s402)。

【0117】このとき、各々の手順に与えるパラメータも(たとえば、G1が円であれば中心O1と半径r、G2が四角形であれば中心O2と二辺の長さ)をそれぞれ送信する。

【0118】この結果、G1の生成手順から相対位置データD1=〔IO1、d1、Q1、KIND1〕、G2の生成手順からD2=〔IO2、d2、Q2、KIND2〕が返信される(s403)。

【0119】これら二つの相対位置データD1、D2は、空間点Pに対する内外判定データ(IO)に応じて次のように場合分けされ、空間点Pに対する和図形の相対位置データが決定される。

【0120】図12に、円と四角形による和図形の説明図を示し、これを参照しながら和の場合分け手順を説明する。

【0121】(1) IO1=IO2="OUT" の場合
これは、図12の空間点P1(121)に相当する。この場合は、相対位置データD1の距離d1と相対位置データD2の距離d2を比較し(s4041)、距離dの小さい方を空間点Pに対する相対位置データDDとする。

【0122】(2) IO1="IN"、IO2="OUT" の場合

これは、空間点P2(122)に相当する。この場合は、図形G1(円)の相対位置データの種別を判定し(s4042)、第一種の場合にその近地点Q1を空間点として図形G2(四角形)の相対位置生成手段に送り(s40421)、その相対位置データD01=〔IO1、d12、Q12、KIND12〕を受信する(s40422)。その内外判定(s40423)でQ1がG2の外にある場合は、空間点P2に対する和図形の相対位置データはDD=D1とし(s404231)、それ以外の場合にはDD=("IN"、d1、nil、

*【0113】反転図形の表示処理は、図形の内外が反転して実領域が元図形と反対になるだけで、基本的には上述した図形Gの表示処理と変わるところがない。

【0114】次に、合成処理の一例として、二つの図形の和(+)図形の相対位置データを生成する機能及びその表示処理について説明する。

【0115】図11は、和図形の相対位置データを生成する手順の流れ図である。和図形の相対位置データ生成手順s400は、例えば表示処理手順s300から、オペランドとして二つの図形名G1、G2を記述した(様式7)のメッセージを受信して実行される。

【0116】

… (様式7)

2) とする(s404232)。

【0123】(3) IO1="OUT"、IO2="IN" の場合

これは、空間点P3(123)に相当する。この場合は(2)と同様にして(s40431~s40433)、空間点P3に対する図形G2(四角形)の近地点Q2が図形G1の外にある場合には(s64331)、空間点P3に対する和図形の相対位置データはDD=D2とし(s404331)、それ以外の場合にはDD=("IN"、d2、nil、2)とする(s404332)。

【0124】(4) IO1=IO2="IN" の場合
これは、空間点P4(124)に相当する。この場合、空間点Pに対する和図形の相対位置データは、DD=("IN"、max(d1、d2)、nil、2)とする(s4044)。

【0125】相対位置データDDは、上記(1)~

(4)の場合分け処理により求められ、空間点Pの送信元、たとえば表示処理手順s300に返信され(s405)、以下のように表示される。

【0126】図13は、和図形の表示処理の説明図と、和の場合分け処理により求められる和図形の相対位置データの例である。

【0127】表示処理手順s300に、(開始点P0、視線V=view及び図形名C)が与えられると、手順s300から図形C相対位置生成手順s400に開始点P0が与えられ、さらに生成手順s400から円Aの相対位置生成手順s200及び説明を省略して図示のない四角形の相対位置生成手順に対して、空間点P0とパラメータが与えられる。

【0128】同図(b)のように、開始点P0に対する円Aの相対位置データD1=(OUT, c0, Q0, 1)と、四角形BのD2=(OUT, s0, R0, 1)が生成手順s400に返信される。手順s400は場合分けにより、両方がOUT及びc0>s0から、和図形Cに対する点P0の相対位置データDD=(OUT, s0, R0, 1)を求め、表示処理手順s300に返信する。

【0129】手順s300は、再び視線V上の次の探索点P1を手順s400に送り、上記処理を行って点P1の相対位置データDD(=D2)を受け取る。これを図示のように繰返し、探索点Pnに対する和図形Cの相対位置データDDが、 $IO=IN$ に変わったところで、点Pn-1と点Pnの座標の二分法から視線V上の和図形の境界点δ1を求める。なお、探索点Pnは図形Aと図形Bの両方に"IN"となるので、上記した和の場合分けの(4)に該当し、相対位置データDDは第二種として生成されている。

【0130】さらに、視線Vを走査幅Wづつ移動しながら、視線方向(view)から見える和図形Cの境界点を求め、その境界点列QQを結んで図形Cが表示できることは、基本図形の表示の場合と同じである。

【0131】次に、本実施例による拡大処理について、相対位置データを生成する機能及び表示処理を説明する。

【0132】図14は、拡大処理の具体例を示し、同図(a)の元図形G(図12に示した円と四角形の和図形)を、同図(c)のように距離rだけオフセットした拡大図形G*を生成、表示する例である。なお、本実施例で言う「拡大」は、オフセット図形を含む広義の意味*

(拡大; 距離r; 図形G)

$G* = \{\text{拡大; 距離}r; \text{図形}G\}$

拡大図形の生成手順s500は、メッセージ制御手段11を介して(様式8)または(様式9)のメッセージを受けると、元の図形Gに対して空間点Pを指定したメッセージを送り(s502)、図形Gの相対位置生成手順から図形Gに対する空間点Pの相対位置データD=[IO、d、Q、KIND]を受けとる(s503)。

【0137】次に、以下のように拡大の場合分け処理(s504)が行われ、拡大図形G*に対する空間点Pの相対位置データDDが求められる。図14に示した空間点P141、P142及びP143を例に説明する。

【0138】(1) $IO="IN"$ の場合

これは、空間点Pが図形G内にあるP1(141)のケースに相当する。受信した相対位置データDのKINDを判定し(s5041)、第一種の場合は、その近地点Qを中心とする半径rの操作円g(Q)を対応させ、g(Q)と半直線PQの交点Rを求める(s50411)。

【0139】次に、交点Rをメッセージとして図形Gに送る(s50412)。その返信データDR=[IO1、d1、Q1、KIND1]を受信し(s50413)、距離d1=rか否かを判定する(s50414)。d1=rの場合は、拡大図形G*の相対位置データDD=["IN"、d+r、R、1]とする。この交点Rは拡大図形G*の境界上にある。d1≠rの場合は、DD=["IN"、d+r、nil、2]とする。

【0140】s5041で、KINDが第二種の場合は、DD=["IN"、d+r、nil、2]とする。

*に用いている。

【0133】拡大処理は、操作図形方式と呼ぶ新規な方法による。即ち、同図(b)に示すパラメータrで中心Qの点対象操作図形g(Q)を介して、前記拡大図形の境界の相対位置データDDを生成する。言い替えれば、点対象操作図形g(Q)を元図形Gの境界上の近地点Qと一致するように対応させ、元図形Gの全体または指定範囲(オフセットの場合は、拡大する境界の一部を指定することも多い)に渡って操作図形g(Q)の集合和をとるものである。

【0134】本実施例の操作図形g(Q)は、半径距離rの円図形を用いて説明する。操作図形g(Q)としては楕円や四角形など、基本的には全ての点対称図形に拡張可能である。

【0135】図15は、拡大図形の相対位置データを生成する手順の流れ図である。拡大図形の相対位置データ生成手順s500は、例えば表示処理手順s300あるいは応用図形処理(NC情報)などから、(様式8)のメッセージを受信して実行される。なお、拡大図形名G*を設定して(様式9)のメッセージとすると、図形名が直接に利用できるため便利である。

【0136】

…(様式8)

…(様式9)

【0141】(2) $IO="OUT"$ & $d \geq r$ の場合
これは、空間点Pが図形Gの外で且つ距離dが距離r以上となるP2(142)のケースに相当する。相対位置データDが第一種の場合は、相対位置データDD=["OUT"、d-r、R、1]とする(s50421)。一方、第二種の場合には、DD=["OUT"、d-r、nil、2]とする(s50422)。

【0142】なお、場合分け(1')と同様にして、操作円g(Q)を対応させ、g(Q)と半直線PQの交点Rを求める処理の記載は省略している。

【0143】(3) $IO="OUT"$ & $d < r$ の場合
これは、空間点Pが図形Gの外で且つ距離がr未満となるP3(143)のケースに相当する。相対位置データDの種別を判定し(s5043)、第一種の場合には、操作円g(Q)を対応させ、g(Q)と半直線PQの交点Rを求める(s50431)。

【0144】次に、交点Rをメッセージとして図形Gに送り(s50432)、その返信データDR=[IO1、d1、Q1、KIND1]を受信する(s50433)。d1=rの場合は、相対位置データをDD=["IN"、r-d、R、1]とする(s50434)。この点RはG*の境界上にあり、Pからの距離は(r-d)となる。また、d1≠rの場合には、DD=["IN"、r-d、nil、2]とする。

【0145】一方、s5043で、相対位置データDが第二種の場合には、上記した操作図形方式を、そのまま

適用することができない。即ち、拡大図形G*に対する空間点Pの内外判定ができず、この点における相対位置データDDを求めることができないからである。このため、本実施例では以下の超球判定方法適用し、空間点Pの内外判定データIOPを求め(s50435)、これを用いて、空間点Pの拡大図形G*に対する相対位置データDD=(IOP, dP, nil, 2)を決定する(s50436)。

【0146】図16に、超球判定法の説明図を示す。超球判定法では、対象図形と同じ空間次元を持つ超球によって判定するが、本実施例では二次元として半径rの円(他の円と区別するために、“超円”と呼ぶことにする)を用いる。なお、超円と上記の操作円は同形で、どちらも図形Gへ対応させるが、対応位置が相違する。超球判定法には、図16の(a)、(b)および(c)に三種類の方法がある。

【0147】(a)の方法では、上記の空間点Pである点P161を中心として、半径rの超円160を描く。円領域162(162aと162bからなる)と図形Gの交差をチェックし、交差する場合には、相対位置データDDの内外判定を“IN”とし、そうでない場合には、“OUT”とする。

【0148】点P161は、円領域162bで図形Gと交差するので、内外判定は“IN”となる。一方、点P163は、超円164と図形Gが交差しないので、“OUT”である。

【0149】(b)の方法では、円領域の変わりに円周を用いることを除けば、(a)と全く同じである。

【0150】(c)の方法では、(b)の方法を効率化するためにモンテカルロ法を適用したものである。即ち、あらかじめ一定数Nを設定し、中心点P、半径rの円周上に最大N個の点を一様乱数を用いて発生させ、逐一、図形Gに対する内外判定を行なう。

【0151】点P161の場合のように、円周162a+162b上に発生させた点群の中に、162bに属する点が検知された時点で、点Pの属性を“IN”と判定する。また、点P163のように、N個全ての点群が図形Gの外部にあるとき、点Pの属性を“OUT”と判定する。

【0152】図17は、上記(c)による超球判定法の手順を示す流れ図である。超球判定法の手順s600は、図15のステップs50435で起動される。即ち、拡大の手順s500から(空間点P、図形名G、半径r)のメッセージを受信し(s601)、内外判定データIOP=“OUT”、距離dP=∞と初期設定する(s602)。

【0153】次に、モンテカルロ法を最大N回試行する(s603)。一様乱数randomを発生し(s6031)、同図(b)の超円160(中心P、半径r)で、乱数に応じた角度θとその円周上の判定点Tの位置

を計算する(s6032, 3)。この判定点Tを図形Gに送信して(s6034)、図形Gに対する点Tの相対位置データ(IOT, dT, QT, KINDT)を受信する(s6035)。

【0154】次いで、内外判定データIOTをチェックし(s6036)、IOT=INであれば、拡大図形G*に対する空間点Pの内外判定データIOP=INとし、距離dP=dTとして(s60361)、s603のループを抜ける(s60362)。

【0155】一方、s6036でIOT=OUTであれば、空間点Pの距離をdP=min(dP, dT)とし(s60363)、次の乱数による新たな判定点Tによる処理を繰り返す。このよう求められた相対位置データ(IOP, dP, nil, 2)を、s500に返信する(s604=s50436)。

【0156】このように、拡大の場合分け処理で、空間点Pが図形Gに対して外部(IO=“OUT”)、相対位置データの距離d<r、KIND=2(第二種)であるときに、空間点Pの拡大図形G*に対する内外判定を超球判定法により可能にしているので、かかる場合にも相対位置データを生成することができる。

【0157】このような本実施例の拡大図形処理によれば、元図形から拡大(オフセット)距離rを半径とする円の操作図形g(Q)を用い、その中心Qを元図形Gの境界上の近地点Qと一致するように元図形Gの境界に対応させ、その集合和をとることで拡大処理を行っている。

【0158】本実施例の操作図形方式による拡大処理を、引用例に記載の「領域のオフセット(同、第9頁左上欄第4行~第10頁左上欄第12行)」と比べると、拡大の場合分けをシンプルにして処理効率を向上している。これは、本実施例の相対位置データから、距離dの“最短距離上界”という概念を排除していること、及び、超球判定法を採用したことにある。

【0159】さらに、IO=INまたはIO=OUT & d<rの場合に、操作円から第一種の相対位置データを返信があっても、その距離d≠rの条件では近地点が決定できず、拡大図形G*の相対位置データDDが第二種となる分岐を含むようにして、拡大処理を正確なものにしている。

【0160】しかも、引用例が特別な条件では結局のところ真の距離を求める複雑な例外処理を必要としている場合にも、本実施例では超球判定法を適用することで、例外なく相対位置データを用いる拡大処理が可能になる。この結果、基本的にはどのような図形処理にも適用が可能となつて、信頼性と使い勝手を大きく向上している。

【0161】図18(a)は、拡大図形の表示処理の説明図、(b)は元図形G(この例では図形C)と拡大図形G*(この例では図形D)の相対位置データテーブルである。

【0162】開始点P0、視線V=view及び図形名Dのメッセージを受信した表示処理手順s300は、拡大図形Dに開始点P0を送信する。拡大図形Dの生成手順s500は、図形Cに開始点P0を送りその手順s400から相対位置データDcを求める。手順s400からの相対位置データDcを、手順s500で拡大の場合分け毎に、操作図形方式で処理して相対位置データDDを求め、表示処理手順s300に返信する。

【0163】同図(b)のように、開始点P0に対する図形Cの相対位置データは、 $Dc = (OUT, d0, Q0, 1)$ となる。P0は図形Cの外で $d0 > r$ であるから、P0に対する拡大図形Dの相対位置データDD = $(OUT, d0 - r, R0, 2)$ を、表示処理手順s300に返信する。

【0164】手順s300は、再び視線V上の次の探索点P1を手順s500に送り、上記処理を繰返し、空間点Pmと空間点Pnの間の境界点δを算出する。なお、相対位置データDcは全て第一種で求まり、本実施例による図形処理では大抵のケースがそうなるので処理効率が低い。

【0165】図19は、拡大図形の表示処理例を示し、相対位置データDcが第二種で且つ、超球判定法を利用する場合である。元図形が楕円の場合には、任意の空間点からそこに至る最短距離を求めることは簡単ではないので、第二種の図形として扱うのがよい。

【0166】同図(a)は、最短距離以下の近似値(距離下界)として第二種の距離データdを算出するときの説明図である。楕円Cの中心Oと空間点Pの半直線POの交点Sとすると、距離 $di = (PiSi / OSi) \cdot b$ となり、楕円Cに対する空間点Pの相対位置データは $Dc = (OUT, di, nil, 2)$ として求まる。ただし、bは楕円Cの短径、Q1、Q2は近地点、dmin1、dmin2は最短距離で、上記の関係から $di < dmini$ が成立する。

【0167】この相対位置データDcを受信して、楕円Cを距離r拡大した図形Dを同図(b)のように表示する。同図(c)はその相対位置データで、 $di < r$ となる点P1、Pnなどで、超球判定法が適用されている。

【0168】以上の表示処理の説明は、全て視線式表示処理方法によっている。この表示方法では、図形の形状が複雑になった場合に、設定される視線方向(view)によって見えるところ(境界の求まる)が異なることがある。かかる場合、探索深さLを図形全域を含む範囲に広げて、一の視線上で複数の境界点を取得する。

【0169】または、複数の視線方向(view; 水平、垂直、45°など)による処理を行って境界点列を取得する。この場合は、取得された境界点に境界線として結べないものがある場合に、視線方向を更に小刻みにした処理を繰り返す必要がある。このため、図形オブジェクトの生成時に表示処理を試行して、予め必要な視線

方向を設定しておく。

【0170】次に、他の表示処理方法の実施例について説明する。これは、引用例の四分木式表示方法を改良した新規な方法で、四分木式パターン表示方法と呼ぶことにする。本実施例の四分木式パターン表示方法は、どのような図形に対しても只1回の一連処理で、相対位置データによる表示が可能になる。

【0171】図20は、四分木式表示処理の概念を説明する模式図である。図形G(例では円A)を完全に含む正方領域Rを設定し、それを中心から4分割して正方領域R1~R4を設定し、図形Gに対する各領域の内外関係を判定し、各々に"内セル"、"外セル"または、"未定セル"のフラグを立てる。

【0172】図21に、正方領域とその内外関係判定の概念を説明する。まず、図形Gに対し任意の空間点Pを中心とする正方領域200を設定する。次に、中心が点Pで半径dの円領域201を設定する。但し、図形Gにする点Pの相対位置データを (IO, d, Q, KIN, D) とする。また、正方領域200の一边が、予め定め

た領域感度δより小さくなるときは、一边をδにする。【0173】同図(a)のように、 $IO = IN$ で且つ円領域201が正方領域200を含めば、この正方領域は図形Gの一部で"内セル"である。円領域201が正方領域200を完全には含まないときは"未定セル"である。同図(b)のように、 $IO = OUT$ であれば、この正方領域は図形Gに含まれず"外セル"である。

【0174】図20で、"内セル"となる正方領域Rは黒色(塗り潰し)、"外セル"は白色、"未定セル"は灰色(まだらの点々)で表している。この未定セルに、図形境界δAの一部が含まれる。

【0175】同図(a)の第一象限~第四象限は、それぞれ分割の第1段階~第4段階に対応して示している。本例では第3段階(第三象限)の未定セルは、第4段階(第四象限)の分割で領域感度δとなって分割を終了する。

【0176】図22は、第四象限における正方領域の分布とその内外判定セル及び表示曲線を示している。未定セルとなる正方領域(イ)~(ト)の四頂点について、○印はその頂点の相対位置データの $IO = IN$ 、●印は $IO = OUT$ の属性を示している。

【0177】図23(a)は、未定セルの4頂点の内外属性のパターンと、対応する表示セグメントを示したものである。内外属性のパターンは、パターン1~パターン6に区分され、パターン4はさらに二つに区分される。各々のパターンに対応して図示の表示セグメントが記憶装置2に格納されている。同図(b)は、パターン41と対応するセル(チ)、パターン42と対応するセル(リ)を示している。

【0178】図24は、四分木式パターン表示の処理手順を示す流れ図である。パターン表示の手順s700

は、(様式10)によるメッセージを受信する(s701)。
* [0179]

[表示; パラメータ(正方領域の中心O, 幅W, 領域感度 δ); 図形名G]

…(様式10)

開始点P=点O、セル幅 $w=W$ を初期設定し(s702)、四分木法処理手順(s703)をP, w, Gをパラメータとして実行する。四分木法処理手順は、s7031~s7037をその内容としている。

[0180] まず、図形Gの手順に点Pをメッセージとして送信し(s7031)、図形Gに対する点Pの相対位置データ(I O, d, Q, K I N D)を受信する(s7032)。ここで、点Pを中心とし一辺が w の正方領域Rにおける点Pと頂点の線分 $=w/\sqrt{2}$ と、相対位置データの距離 d を比較し(s7033)、後者が大($d > w/\sqrt{2}$)のときリターンする。この場合は、当該領域が図形Gの境界に関係しないからである。リターン先は、当該処理を呼び出していた四分木法(s703/s7037のi)である。

[0181] 一方、 $d \leq w/\sqrt{2}$ のときは、セル幅 w が領域感度 δ 以下になったか判定し(s3074)、領域感度 δ 以下のセルについて4頂点P1~P4の相対位置データを受けとって、4頂点P1~P4の図形Gに対する内外属性を判定する(s30741)。

[0182] 各セルの4頂点の内外属性の組合せパターンを分類し(s30742)、該当するパターンのセグメントを表示し(s30743)、本処理の呼出し先にリターンする(s30744)。

[0183] s3074で、 $w > \delta$ であれば、同図(b)のように $w'' = w/2$ を求め(s7035)、点Pを中心として一辺 w'' の正方領域の四頂点Pi(i=1~4)を求める(s7036)。そして、(Pi, w'' , G)をメッセージとする四分木法処理手順を、i=1~4まで繰り返す(s7037)。

[0184] 本実施例によれば、相対位置データを利用するどのような複雑な図形表示においても、上記領域感度 δ までの一連の四分木処理によって正確に表示でき、しかも最小セルの4頂点の内外属性の組合せパターンに応じて、直ちに表示セグメントが選択され、引用例の四分木法表示処理に比べ、その処理精度と速度を大幅に向上している。

[0185] なお、四分木法は、ハンター(G. Hunter)によって提案されたものであり、文献等に周知である。

[0186] 以上、基本図形及びオペランド図形の主なものについて、その相対位置データの生成方法と表示処理を説明した。ところで、上記の図形処理において、上層の図形の相対位置データ生成手段は、下層の図形の相対位置データ生成手段とメッセージを送受信しながら、所望の相対位置データを生成している。このメッセージの送受信は、通信制御機能11によって処理される。

[0187] 図25に、図形の表示処理を例にした通信処理の流れを示す。上述のように、円図形Aと四角形Bを合成した和図形Cを拡大して生成される拡大図形Dを例にとる。

[0188] 表示処理手段(s300)191は、通信制御機能11を介してメッセージS1(表示; 表示パラメータ; 図形D)を受信すると、視線方向(view)を設定し探索点Pを拡大図形DにメッセージS2を送信する。正しくは、(P; 図形名D)のメッセージを通信制御機能11に送信し、通信制御機能11から図形名Dのオブジェクトデータ(# o b d)を読み出し、その手順ポインタ(# z z z)にある拡大図形相対位置生成手順192を起動して、メッセージS2(点Pとオペランド図形Cのポインタ(# o b +))を渡す。

[0189] 以下同様にして、拡大図形生成手順192はS2を受信すると点PをメッセージS3を図形Cの和図形相対位置生成手順193に送信し、さらに図形A及び図形Bそれぞれの相対位置生成手順194, 195にメッセージS4, S5で点Pが送られる。

[0190] これにより、円生成手順194及び四角形生成手順195は、それぞれ点Pの相対位置データDa, Dbを求めて生成手順193に通信制御機能11を経由して送信する。和図形生成手順193は、受信した相対位置データDa, Dbについて和の場合分けをして求めた相対位置データDcを拡大図形生成手順191に送信し、さらに拡大の場合分けにより求められた拡大図形Dに対する点Pの相対位置データDdが、表示処理手段191に返信される。表示処理手段191は、順次探索点Pをメッセージして、上記通信処理を経て図形Dの境界点列QQを求め、表示装置4に出力する。

[0191] 本実施例ではこのように通信制御すること、上層の図形オブジェクトデータはその生成手順ポインタと下層の図形ポインタを記憶するのみで、上層と下層の間の再帰的な処理が円滑に且つ効率良く実行できる。

[0192] 次に、応用図形処理の実施例として、面積計算とフィレットを説明する。

[0193] 図26は、面積計算の概念を模式的に示した説明図である。面積計算は、上述の四分木法を用いて行われる。まず、図形Gを完全に囲む正方領域200を設定し、その中心を通り互いに直交する二つの線分203, 204により、全体を四つの正方領域210~240に分割し、領域200に対する四分木構造として記憶する(第1段階)。

[0194] 次に、正方領域210~240の各々に対して、図21を用いて説明したのと同様の内外判定処理

を行い、正方領域毎に、“内部”、“外部”または、“未定”のフラグを立てる。上記で“内部”となる正方領域は、ツリー構造の図形オブジェクトの葉の部分（基本図形）に存在している。

【0195】上記で“未定”となる正方領域が、ツリー構造の葉の部分に存在している場合は、その正方領域を再び四つの正方領域に分割し（第2段階）、その内外判定を行う。正方領域が“外部”の場合または“未定”で葉の部分でないとき、当該領域の分割操作を終了する。

【0196】同図には、この分割操作の第1段階～第4段階までの分割状況を示し、第4段階で正方領域の一边の長さが領域感度 δ 以下となり分割を終了している。図示の正方領域（イ）は第2段階でその内外判定は“外部”、正方領域（ロ）は第3段階でその内外判定は“内部”、正方領域（ハ）は第4段階でその内外判定は“未定”となる。

【0197】第4段階、即ち領域感度 δ 以下の場合の未定領域は、その中心Pの内外判定によって、“内部”または“外部”とする。最後に、上記の操作で得られた四分木構造をスキャンして、“内部”の属性を持つ正方領域の面積の総和をとる。

【0198】図27に、面積計算手順の流れ図を示す。応用図形処理機能15は、面積計算のメッセージ（面積；パラメータ；計算対象となる図形G）を受けると、面積計算処理手順s800を呼び出し、対称の図形名Gとパラメータ（正方領域の中心O、幅W、領域感度 δ ）を受信する（s801）。そして、開始点P=点O、セル幅 $w=W$ 、面積 $S=0$ を初期設定し（s802）、P、w、Gをメッセージして四分木法の処理を開始する（s803）。

【0199】四分木法は、まず点Pをメッセージとして図形Gに送信し（s8031）、相対位置データ（I、O、d、Q、KIND）を受信する（s8032）。そして、 $d > w/\sqrt{2}$ または $\delta \geq w$ であるか判定し（s8033）、そうであれば、IO=“IN”かを判定し（s80331）、そであれば、“内部”、正方領域の面積の累積計算である $S=S+w \times w$ を行って（s803311）、呼出し先にリターンする（s803312）。それ以外の場合は、何もしないでリターンする（s803313）。

【0200】ステップs8033で、判定が否の場合は $w''=w/2$ として（s8034）、点Pを中心として、一边 w'' の正方形の四頂点P1、P2、P3、P4を同図（b）の陽に求める（s8035）。そして、四分木法にメッセージ（Pi、w''、G）して、上記s8031～s8035を $i=1 \sim 4$ まで、繰り返し処理する（s8036）。

【0201】本実施例によれば、境界が陽に得られないような複雑な図形Gの面積を、図形Gに対する空間点Pの近傍領域の内外関係を、相対値データを用いて判定す

ることで、簡単且つ迅速に計算できる。

【0202】以上は、面積計算の例であるが、ここで示した方法の本質的な部分、即ち、与えられた図形に対して、任意の空間点Pの近傍領域の内外判定を行なう手法は、そのまま干渉チェックの手順などに適用可能である。

【0203】次に、応用図形処理の他の例として、図28を参照してフィレット処理を説明する。フィレットは、同図（a）の図形領域Gの凸部を半径 $\delta 1$ 、凹部を半径 $\delta 2$ で丸めて、同図（c）の新たな図形領域G*を作る処理である。

【0204】応用図形処理機能15は、フィレットのメッセージ（フィレット；パラメータ（ $\delta 1$ 、 $\delta 2$ ）；対象図形名G）を受けると、フィレット処理手順263を呼び出して次の処理を行う。

【0205】[フィレット； $\delta 1$ ； $\delta 2$ ；図形G]= [オフセット； $\delta 1$ （反転；（オフセット； $\delta 1 + \delta 2$ ；（反転； $\delta 2$ ；図形G））））を行う。図29に示す各図形の生成過程と処理関係を具体的に示すと、以下のようになる。

【0206】 $G_1 = [\text{オフセット；}\delta 2；\text{図形G}] \Rightarrow G_2 = [\text{オフセット；}\delta 1 + \delta 2；\text{反転図形G}_1] \Rightarrow G_3 = [\text{反転；図形G}_2] \Rightarrow G^* = [\text{反転；}\delta 1；\text{図形G}_3]$ となる。なお、オフセット処理は、上記した本実施例の拡大処理で行われることは言うまでもない。

【0207】以上に説明した各種の図形生成処理における相対位置データ生成手順は、空間点Pが与えられると、点Pと対象形状Gとの位置関係に関して詳細計算を行い、相対位置データを生成するものであった。この相対位置データの生成は本発明の実施に必須の基本手順であり、その処理性が高いほど実用上のメリットは大きく、適用範囲も広いものとなる。

【0208】次に、この相対位置データの計算をより簡略化し、相対位置データの生成手順を高速化する領域データ方式について説明する。この方式は、図形形状に関する参照データである領域データを図形オブジェクトデータに追加し、これを基に相対位置データの簡略計算を行なうもので、静的領域データと動的領域データを用いる二つの方法がある。

【0209】図29は、静的領域データを用いる相対位置データ簡略計算の概念を示す説明図である。

【0210】静的領域データを用いる方法は、図形Gの境界 δG を内部に含む極小、または近似的に極小で且つ単純な領域図形Cg、例えば、円または四角形を図形Gの生成時に求め、あらかじめ図形オブジェクトデータ（様式2）のパラメータに含める。領域図形に円を用いることにすると、領域円Cgは図形Gに関して定まり、様式11に示す形式で、領域データが与えられる。

【0211】

ここで、円C_gの内外属性は、領域円C_gが同図(a 1)のように図形Gの内部を含むときはOUTで、同図(a 2)のように図形Gの外部を含むときはINを取るものとする。

【0212】静的領域データによる相対位置手順は次のように行われる。すなわち、空間点Pがメッセージとして図形Gに送られてきたとすると、まず、図形Gの領域データを参照して、点Pが領域円C_gの内部にあるか、外部にあるかを判定する。外部にある場合には簡略計算が可能であり、点Pと円C_gの距離dを求めて、様式4による相対位置データ(円C_gの内外属性、d、n_i 1、2)を返信すればよい。点Pが円C_gの内部にある場合には、前述の詳細な相対位置データ生成手順にしたがうものとする。このように、点Pが図形Gの境界δGから十分離れているときには、図形Gを領域円C_gで代替した簡略計算によって、相対位置データの生成処理を*

〔点P_iの内外判定データ、空間点P_i、距離データd_i〕… (様式12)

この領域データは、メッセージが初めて送られてきた空間点P₀における〔点P₀の内外判定データ、空間点P₀、距離データd₀〕を基に、以下のように更新しながら利用される。

【0216】点P₀における領域データの設定後、図形Gが次の点P₁をメッセージとして受けたとすると。このとき、まず、図形Gの領域データを参照し、点P₁が点P₀を中心とし、半径d₀の円の内部にある場合、簡略計算が可能であり、点P₁を中心とし、円C₀に外接する円C₁の半径d₁を求め、(点P₀の内外判定データ、d₁、n_i 1、2)を点P₁の相対位置データとして返信する。この場合には、領域データの更新はない。

【0217】なお、点P₀から点P₁までの移動距離s₀₁を利用し、距離データd₁=d₀-s₀₁と簡略計算される。図形表示などの応用処理に於けるように、一つ前の点の近傍に関係付けられて次の点が決められる、所謂コヒーレントなメッセージ点列が与えられる場合は、更新された領域データの距離データd_iを基に順次、各点の距離データを簡略計算できるので、相対位置手順の高速化が達成できる。

【0218】もし、次のメッセージが点P₂のように、円C₀の外部になるときは、点P₂に対して簡略計算を行わず、通常の詳細計算により得られた相対位置データに基づき、領域データを(点P₂の内外判定データ、点P₂の座標、距離データd₂)として更新する。

【0219】以上のように、初期または処理中に設定される領域データは、各図形オブジェクトデータに付加した形で記憶される。図31(a)に、円H₁、四角形H₂および三角形H₃の和集合からなる図形W₂のトリート構造を、同図(b)に、その図形オブジェクトデータ構造を示している。図示のように、各図形オブジェクトデータに付加した少量の領域データは、トリート構造の各ノ

*高速化できる。

【0213】図30は、動的領域データを用いる相対位置データ簡略計算の概念を示す説明図である。動的領域データによる相対位置手順は次のように行われる。これは、全てにおいて、静的領域データの場合と対照的である。動的領域データは、静的領域データのように図形生成時に設定されるものではなく、応用処理手順の適用中に設定され、しばしば更新されるデータである。

【0214】すなわち、図30(b 1)、(b 2)に示すように、対象とする図形Gに対して、メッセージ空間点P_iが送られてきたときに、その点の相対位置データを計算後、そのデータの一部を様式12に示す形式で、領域データとして図形Gのオブジェクトデータに登録する。

【0215】

ードに対応する図形毎に同一の形式で記憶される。すなわち、静的領域データ(様式11)と動的領域データ(様式12)のデータ形式は同一であり、相対位置データの生成処理上では同じものとして扱える。

【0220】このような静的領域データと動的領域データは、その一方または両方を領域データとして用いることによって、相対位置手順の高速化が可能になる。簡略計算によって、距離データの精度は多少落ちるが、計算手順が点と円の距離、言い換えると2点間の距離の計算に帰着され、非常に簡単になるからである。後述するように、相対位置データの計算に必要なトリートの探索において、領域データの採用によるトリートの枝切りが可能になるので、探索の高速化も併せて達成される。以下、領域データを用いるオブジェクト図形の相対位置データ生成方法を詳細に説明する。

【0221】図32は、和図形W₂のトリート構造における静的領域データの求め方を示す説明図である。静的領域データの生成は、図形オブジェクト生成時に一回だけ行なわれるが、各種の図形オブジェクトに対応した体系的な手順によって達成される。以下、各種の図形処理に必須となる四種類の図形(基本図形、反転図形、和図形及び拡大図形)における領域データの生成手順を説明する。

【0222】基本図形の領域データは、図32(a 1)、(a 2)、(a 3)に示すように、円H₁、四角形H₂または、三角形H₃の場合、図形オブジェクト固有のパラメータから、その外接円を求めることにより、領域円C₁、C₂またはC₃がそれぞれ決まる。各領域円の内外属性はOUTである。

【0223】反転図形の領域データは、既に作成済みのオペランド図形の領域データを取り出し、その内外属性を反転することによって得られる。例えば、同図(a

3) の図形H3の反転図形は、同図(b1)のHC3となるが、この図形の領域円CC3は、図形H3の領域円C3と等しく、単に、内外属性が反転したものになる。反対に、図形HC3を反転すると図形H3になるが、この場合にも上記の関係が成立している。

【0224】和図形の領域データは、二つのオペランド図形を取るため、まず、これらの図形に設定されている領域データを取り出す。この結果、各領域円に設定されている内外属性の組み合わせによって、図32(c1), (c2), (c3)に示すように三つの場合に分

10 かれる。
【0225】図32(c1)のように、オペランド図形H2およびH3の内外属性が共にOUTの場合、二つの領域円C1及びC2に内接する円C+が、和図形の領域円であり、その内外属性はOUTである。

【0226】図32(c2)のように、一方のオペランド図形HC3の内外属性がINで、他方の図形C2の内外属性がOUTの場合、H2の領域円C2がHC3の領域円CC3を内部に含まないときには、内外属性がIN

20 となる図形HC3の領域円CC3を和図形H+の領域円C+とする。
【0227】図32(c3)のように、図形HC3と図形H2の反転図形(図形HC2、領域円CC2とする)の二つの内外属性が共にINの場合、二つの領域円CC2とCC3の共通弦を直径とする円C+を和図形H+の領域円とする。円C+の内外属性はINである。なお、前記共通弦が存在しない場合、即ち、二つの領域円が交差しない場合には、図形H+は全空間を意味するトリビアルなケースとなるので、外部属性を有する半径0のダミーな円を領域円とする。

【0228】拡大図形の領域データは、拡大(様式8)のパラメータである半径 r と、オペランド図形の領域データを取り出す。領域円の内外属性によって、図32(d1), (d2)に示すように二つの場合に分かれる。内外属性がOUTの場合、図(d1)に示すように、図形H3の領域円C3の半径を r だけ延長した、円CEを拡大図形HEの領域円とする。内外属性がINの場合、図(d2)に示すように、図形HC3の領域円C3の半径を r だけ短縮した、円CEを拡大図形HEの領域円とする。但し、パラメータ r が円CC3の半径より大きい場合には、円CEの半径を0とする。即ち、図形HEが全空間を意味するトリビアルなケースである。

【0229】図33に、静的領域データを用いた和図形W2の相対位置データ生成の概念と、各点の相対位置データのテーブルを示す。

【0230】同図(a)は、和図形W2として組み立てられた図形H1、H2及びH3が配置された状態と、これらの領域円C1、C2及びC3、和図形W1、W2の領域円CW1、CW2を図示している。さらに、ベクトルLの向きに移動する物体(Bと呼ぶ)に関する代表点

の時系列位置 $\{P_i, i=1, 7\}$ を示している。その他の記号については、図中に説明している通りである。本実施例は、上記の時系列点に対して、時系列順にメッセージを図形W2に送って、物体Bと図形H1、H2またはH3との干渉チェックを行なうことを想定している。

【0231】同図(b)は、静的領域データによる簡略計算を含む各空間点の相対位置データを具体的に示したものである。あらかじめ、各図形に各々の領域データが設定されている。テーブルの空白部は計算の省略による

10 トリー探索の枝切り(計算なし)、斜線部は通常の詳細計算、斜線部以外は領域データを用いた簡略計算をそれぞれ示している。
【0232】空間点P1、P2は和図形W2の領域円CW2の外部にあるので、領域円CW2に対する空間点P1、P2の相対位置データを返信するのみでよく、簡略計算となり、その他の図形に対する計算は全て省略されている。空間点P3、P4はCW2の内部でC1及びCW1の外部にあるので、W2に対しては詳細計算、H1及びW1に対しては簡略計算を行ない、図形H2とH3

20 に関しては省略される。なお、和図形の距離データ d_i は前述したように、短い方 $d_{3w1} (< d_{3c1})$ が採用される。空間点P5、P6はCW1の内部でC2及びC3の外部にあるので、図形W2及びW1は詳細計算、他は簡略計算を行う。空間点P7は領域円C3の内部にあるので、図形H1及びH2が簡略計算、他は詳細計算となる。

【0233】このように、空間点 P_i が領域円の外になる場合に、

30 トリー探索の枝切りや簡略計算が頻繁に行なわれていることが分かる。従って、図示を省略している空間点P7の後、領域円CW2の外部に出たからは、全ての空間点に対してH1以降の探索の枝切りが行なわれる。なお、静的領域データによる手順では、メッセージの順序に依存しない。

【0234】図34に、動的領域データを用いた和図形W2の相対位置データ生成の概念と、相対位置データのテーブルを示す。この実施例における状況設定は、図33の静的領域データの場合と同様である。但し、領域円は存在しない。同図(b)に示すように、各点の各図形に対するデータは二段になっていて、上段が相対位置データ、下段が動的に更新される領域データの内容を表す。図示で、相対位置データの詳細計算が行われる場合には領域データの更新が行われ、簡略計算の場合には領域データは更新されない。テーブルの空白部は計算の省略による

40 トリー探索の枝切りを示している。
【0235】まず、初めてのメッセージとして空間点P1が与えられると、通常の相対位置手順による詳細計算が図形W2に関する各図形に対して行われ、様式12による領域データ(OUT, P1, d1h1)が生成される。空間点P2では、前述のように時系列点のコヒーレ

ント性を利用して、相対位置データ (OUT, $d1h1 - s12$, nil, 2) を簡略計算して返信する。空間点 P2, P3 では、H1 以降の図形の計算は省略される。

【0236】空間点 P4 で半径 $d1h1$ の円の外になると $s14 > d1h1$ 、円 H1 に対する相対位置データが詳細計算され、領域データは (OUT, P4, $d4h1$) と更新される。和図形 W2 を構成する一方の図形 H1 の相対位置データが計算される場合、他方の図形 W1 に対する相対位置データの計算も行う。その逆に、和図形 W2 を構成する図形 W1 と図形 H1 の相対位置データの計算がなされるとき、和図形 W2 の相対位置データが詳細計算されて領域データも更新される。前述のように、和図形の相対位置データは、詳細計算と言っても、実質は一方の図形の相対位置データを転用するに過ぎず、高速処理が可能である。

【0237】ここで、和図形 W1 の相対位置データを与える図形 H2 と当初の点 P1 の距離データ $d1h2$ は、点 P1 から点 P4 の移動距離 $s14$ より依然として大きいので、和図形 W1 に対する相対位置データは簡略計算によって行われる。また、和図形 W2 に対する相対位置データは、 $d4h1 > (d1h2 - s14)$ であることから、和図形 W1 の相対位置データがそのまま採用され、領域データは (OUT, P4, $d1h2 - s14$) と更新される。

【0238】空間点 P5 では、図形 H2 に対する相対位置データの詳細計算と領域データの更新が行われる。一方、図形 H1 に対しては $d4h1 > s45$ 、図形 H3 に対しては $d1h3 > s15$ となるので、簡略計算のみが行われる。図形 H2 に対する詳細計算に応じて和図形 W1、さらに和図形 W2 の詳細計算が行われる。

【0239】空間点 P6 では、和図形 W1 を構成する図 *

〔空間点 P, カットオフ距離 (dcut)〕

このカットオフ距離 dcut は、次の二通りに用いられる。

【0244】一般に、領域データに基づく簡略計算では、簡略計算の結果求めた距離データが、あらかじめ定められた値、すなわち dcut より小さい場合には、簡略計算の結果を捨てて、通常の手順に基づく詳細計算を行なう。このことにより、精度の低下による影響を軽減する。

【0245】さらに、物体と対象図形の間で概略の干渉チェックを行なうために、本実施例の拡張生成手順を用いる場合、点 P を物体の代表点とし、この点を中心とし、物体を包む包絡円の半径を dcut とすれば、拡張手順の結果、得られた相対位置データの内外属性が OUT (IN) で距離データが dcut より大きい場合には、物体は対象形状の外 (内) にあると判断できる。

【0246】以下、図 35 の拡張手順を詳細に説明する。ステップ t101 はメッセージの受信で、t102

* 形 H2 及び H3 は共に簡略計算である。この場合、和図形 W1 は簡略計算によることも可能であるが、本実施例では空間点と対象図形の距離が一定以内になると距離精度を維持するために、領域データも更新している。

【0240】空間点 P7 では、初めに空間点 P1 で求めた各図形の領域データにおいて最長の距離 $d1h3$ を超える距離に設定されている。言い替えば、和図形 W2 内の全ての図形に対する空間点 P1 と P7 の視線方向が逆転している。このような場合、各図形に対する点 P7 の相対位置データを詳細計算し、領域データを更新する。空間点 P7 の後は、全ての空間点に対して H1 以降の探索の枝切りが行なわれる。

【0241】このように、静的領域データに基づく手順と動的領域データに基づく手順は、メッセージ点 P が、対象図形の境界から十分離れているときは、いずれも同様に効果的である。また、前者はメッセージ順序に関係しないのでメッセージ点列にコヒーレントな性質があるとなかろうと影響を受けない。他方、後者は、一般によく現われるコヒーレントなメッセージ点列に対しては、メッセージ点が図形の近くにある場合でも、前者より遥に効果的に作用する。このように、二つの手順の一方、ないし両方の手順を取り入れた相対位置手順は高速化にとって有効な手段となる。

【0242】図 35 は、静的領域データに基づく手順と動的領域データに基づく二つの手順を融合した、相対位置データ生成の拡張手順 (t100) を示すフローチャートである。拡張前の相対位置データ生成手順では、メッセージとして様式 3 に示すように空間点 P が与えられたが、本例の拡張手順では、後述するカットオフ距離 = dcut を追加して、様式 13 によるメッセージを図形 G の相対位置データ生成処理手順へ送信する。

【0243】

… (様式 13)

～t104 は静的領域データに基づく簡略計算を行なう。ここで、t104 の条件が満たされる場合、即ち、「点 P が領域円の外にあり、且つその距離 $d1$ がカットオフ距離 dcut より大きい」と言う第 1 の条件が成立する場合には、ステップ t1041 で相対位置データが返信される。

【0247】この第 1 の条件が成立しない場合には、ここでの結果は無効となり、動的領域データに基づく簡略計算 t105～t107 を行なう。ここで、ステップ t107 の条件が満たされる場合、即ち、「点 P が領域データに含まれる、空間点 O2 を中心とし、半径 $r2$ の円の内にあり、且つその距離 $d2$ がカットオフ距離 dcut より大きい」と言う第 2 の条件が成立する場合には、ステップ t1071 で相対位置データが返信される。

【0248】前記の第 1 及び第 2 の条件が共に成立しない場合には、ここでの結果は無効となり、ステップ t108 において拡張前の手順による詳細計算を行う。拡張

前の手順としては既に、処理図形に応じて基本図形の図5、和図形の図11、拡大図形の図15などを示している。最後のステップt109では、ステップt108の結果である相対位置データDDに基づいて、動的領域データを更新した後、相対位置データDDが返信される。

【0249】上記の位置データ生成の拡張手順においては、二種類の領域データに基づいた手順を示したが、その一方だけの手順を構成することは容易である。例えば、図35のフローから、t102~t104の部分を除けば動的領域データのみによる拡張手順、t105~t107の部分を除けば静的領域データのみによる拡張手順となる。いずれの場合にも、この拡張手順は、大量の物体のレイアウトとか、障害物の中での物体移動における、高速な干渉チェックの方法として極めて有効である。

【0250】次に、本発明の相対位置データによる三次元モデルにおける実施例を説明する。これまでの実施例は、主に二次元モデルを念頭に説明している。しかし、本発明の画像処理は、対象画像が複雑になればなるほど、従来の画像処理に比較して処理性が向上できるので、三次元モデルは好適な対象といえる。

【0251】三次元モデルの場合にも、図1及び図2に示した図形処理装置の各機能や処理手順の基本内容は変わらない。ただし、図2及び図3におけるデータ構造は三次元的な内容に置き代わる。まず、二次元モデルで説明した三次元版について、概略の説明を行う。

【0252】三次元モデルによる図形相対位置生成機能は、二次元の場合と同様に、基本図形、反転図形、和図形及び拡大図形の四つの生成処理に大別される。基本図形に関する手順は、円や四角形の代わりに球や、四角柱に関する手順を備える必要がある。ところが、例えば、球の場合には、図5で示した円に対する手順と形式的には同じであり、単に、点の座標値を三次元で取り扱えばよい。

【0253】図36に、三次元モデルの図形オブジェクト生成機能で取り扱う実用的な基本図形群を示す。四角柱、四角錐体、三角柱、三角錐体、球、円柱、円錐体、トラス、回転楕円体、回転放物体、回転双曲体等の三次元基本立体について、与えられた三次元空間上の任意の点Pに対して、相対位置データを計算する図形生成手順を、前述の球における場合と同様に構成して図形オブジェクト生成手順ファイル23に記憶して管理する。また、これらの生成手順を用いて生成したオペランド図形は、その図形名、アドレス、パラメータなどのデータをオブジェクトデータファイル24に格納する。

【0254】三次元モデルによる反転と和図形の図形生成は、データを三次元で扱うことを除いて、二次元の場合の手順と同様であり、たとえば和図形の場合には図11のフローと同じになる。

【0255】三次元モデルによる拡大図形の生成は、図

15の拡大の手順S500の中で、ステップs50435に対応する超球判定手順S600(図17)が二次元モデルに限られることから、後述する三次元用の超球判定法1(または2)と置換することで、拡大の手順S500が三次元モデルに拡張して使える。

【0256】また、図形相対位置生成機能は、三次元モデルの集合演算による図形生成手段を備え、二次元の場合と同様に、反転や和(+)を組み合わせたブール代数によって、二つの図形をオペランドとする差(-)や積(*)による図形生成を可能にしている。さらに、図形相対位置生成の機能は、三次元モデルの座標変換を行う図形生成手段を備え、生成済みの図形のアフィン変換、移動、回転による図形生成を可能にしている。

【0257】三次元モデルにおけるオフセット図形は、拡大図形における操作図形として球を用いた場合であり、前述の二次元モデルの場合と同様に構成される。これにより、フィレット図形、即ち、与えられた図形の凸部と凹部を、それぞれ半径r1、半径r2で丸めた図形を生成する手順も、後述するようにオフセットと反転の組み合わせで構築できる。

【0258】以上のように、図形オブジェクト生成手段12と相対位置データ生成手段13を中核とする図形相対位置生成機能を、前述の二次元から三次元モデルを含むように拡張しておくことにより、広範囲の三次元の基本図形を基にその反転、和、拡大等のオペランド図形の生成が可能になり、設計や製造あるいは検査等で必要となる図形の配置、変形など種々の応用処理が実行できる。

【0259】図37~図39に、三次元モデルの典型的な一例である金型形状の図形生成手順を説明する。図37は、対象となる金型形状Gの外観図である。この金型形状Gは、図38(a)に示す図形Gのトリート構造に従い、以下の手順によって生成される。

【0260】回転楕円体EL、四角柱RE、球CU、円柱CYの各基本図形、及び、和(+)図形、差(-)図形、フィレット図形の各々の生成手順が、図38(b)に示すように、図形生成手順ファイル23に格納されている。

【0261】まず、様式1によるメッセージ(回転楕円体;中心、長径、短径)により、回転楕円体REの図形オブジェクト生成が指示される。図形オブジェクト生成処理(s110)が起動して回転楕円体の相対位置生成手順(#iii)が呼び出され、様式2により生成した回転楕円体ELのオブジェクトデータは、図39に示すようにファイル24の#obelに格納される。同様に、四角柱RE、球CUの各オブジェクトが生成され、それぞれファイル24の#obre、#obcuに格納される。次に、メッセージ(+;;RE,EL)により、和(+)図形相対位置生成手順(#jjj)が呼び出され、生成された和図形G1のオブジェクトデータが

ファイル24の#obg1に格納される。さらに、メッセージ(+; ; G1, CU)により、和図形G2のオブジェクトが#obg2に格納される。

【0262】ここで、メッセージ(フィレット; 半径=0、半径=r; G2)により、フィレット図形の相対位置生成手順(#111)が呼び出され、和図形G2に対する凹部丸め半径r(凸部は0)による丸め処理が行われ、図形G3として#obg3に格納される。次に、メッセージ(円柱; 中心、軸方向、半径、高さ)により円柱CY1、円柱CY2がそれぞれ生成され、さらに、メッセージ(+; ; G3, CY1)により、和図形G4が生成される。最後に、メッセージ(-; ; G4, CY2)により、差(-)図形の相対位置生成手順(#kkk)が呼び出され、図形G4と円柱CY2の差図形である金型形状Gが生成される。

【0263】これらの処理作業は、二次元モデルの場合と同様に、実際には後述する表示処理を介して、画像表示しながら進められる。また、前述の領域データによる相対位置データ生成手順は、三次元モデルの場合にも可能である。すなわち、静的手順の場合には、領域円の代わりに領域球を用いること、動的手順の場合には点の座標を三次元で扱うことにより、二次元における生成手段がそのまま適用でき、高速処理される。

【0264】三次元モデルにおける、相対位置手順の体系を完結させるためには、超球判定法の三次元版を備える必要がある。二次元モデルにおける超球判定法の一例である図17は、円を用いたモンテカルロ法であるので、三次元の場合には使えない。そこで、三次元超球判定法として、球面(三次元球の境界面)を操作図形とするモンテカルロ法(超球判定法-1と呼ぶ)または球面四分木法(超球判定法-2と呼ぶ)を新たに備える。モンテカルロ法が確率的な方法であるとすれば、球面四分木法は、これとは対照的に決定論的な方法である。

【0265】図40に、球面を操作図形とするモンテカルロ法による三次元超球判定法の実施例を示す。同図(a)に、t200として示す超球判定法-1は、図17の二次元モンテカルロ法を三次元に拡張したものである。メッセージ受信部t201、超球判定法-11(本体部)t202、メッセージ返信部t203からなる。

【0266】本処理の本体部をなす超球判定法-11は、メッセージ点Pを中心とし、半径rの球面と図形Gとの交差の有無を、最大N個の点を球面上にランダムに発生させ、これらの点の図形Gに関する相対位置データを調べることによって判定し、拡大手順のための必要データを抽出するものである。

【0267】超球判定法-11(t202)は、球面上に一樣にランダム点を発生させるために、1点当たり、2個の(random1, 2)一様乱数を発生させてその座標を決める(t20221~t20223)もので、この点で図17の方法と異なり、他は処理が三次元

になるだけで同じである。

【0268】具体的には、モンテカルロ法を最大N回指向する(t2022)。一様乱数randomを発生し(t20221)、同図(b)超円(中心P、半径r)で、乱数に応じた角度 θ 、z座標zzとその球場の判定点Tの位置を計算する(t20222、t20223)。この判定点Tを図形Gに送信して(t202241)、図形Gに対する点Tの相対位置データ(IOT、dT、QT、KINDT)を受信する(t202242)。次いで、内外判定データIOTをチェックし(t202243)、IOT=INであれば拡大図形G*に対する空間点Pの内外判定データIOP=INとし、距離dP=dTとして、t2022のループを抜ける。一方、t202243でIOT=OUTであれば、空間点Pの距離をdP=min(dP, dT)とし、次の乱数による新たな判定点Tによる処理を繰り返す。このように求められた相対位置データ(IOP、dP、nil、2)を、拡大図形の相対位置生成手順(s500)に返信する(t203)。

【0269】次に、球面四分木法による三次元超球判定法の実施例を説明する。球面四分木法は、図41(a)に示すように、点Pを中心とする半径rの球面と、対象となる図形Gとの交差を判定するために、あらかじめ、前記球面を北極N及び南極Sを通る経度 $\pm\pi$ の経線に切り込みを入れて、同図(b)に示すように、矩形領域(球面展開図と称する)に展開し、球面上の点と球面展開図の間に、経度 θ と緯度面zによる座標(θ , z)に関して一対一の関係を付けておき、この球面展開図に四分木法を作用させることによって、探索の絞り込みを行なうことを特徴とする。

【0270】図42は、球面四分木法に基づく超球探索法(超球判定法-2)の手順の説明図である。球面四分木法t300は、t301でメッセージ(空間点P、図形名G、半径r)を受信し、t303で球面展開図またはその四分木領域である矩形領域(同図(b)参照)のデータをセットし、t302で球面四分木法に関する再帰処理(超球判定法-22)を行なって、点Pの相対位置データを求め、t304でこのデータを返信する。

【0271】超球判定法-22は、前記(P、G、r)と、参照点(θ_0 , z0)及び展開図における矩形領域のサイズ(W、H)を受信し、t3031~t3034で参照点T(θ_0 , z0)の図形Gに関する相対位置データを求める。次に、t3035~t3036で前記矩形領域と一対一に対応する球面上の領域(四辺形パッチ)に内接する中心Tの球面半径rrと、ここで計算を打ち切った場合の誤差(error)を求める。t3037で、もし、点Tの内外属性IOTがI-Nであるか、rr<dPか、またはerrorが、あらかじめ設定した打ち切り誤差 ϵ より小さい場合には、この矩形領域に関する探索は終了したものと、t30371で点

45

Pの相対位置データを返信する。即ち、この処理は再帰処理になっているので、呼び出し側の超球判定法-22に戻る。

【0272】上記の条件が成立しない場合には、t3038で前記矩形領域をその中心に関して四分割し、t3039でこれら分割された矩形領域に対してt30391~t30394で、超球判定法-22自身の再帰処理を行なう。即ち、超球判定法-22に各矩形領域に関するメッセージ：(P、G、r)、(θi、zi)及び(W/2、W/2)を送信する。t20392でその結果である相対位置データ(10i, di, nil, 2)受信し、t30393で、もし、10iがINならば、交差が判明したものと、所定のデータをセットしてt3039を抜ける。一方、OUTの場合には、t30394でdPの値を更新する。最後に、t3040で点Pの相対位置データを返信する。

【0273】ところで、本発明の相対位置データによる画像生成手順の中で、最も時間のかかる処理は超球判定法である。これは二次元、三次元を問わず、また、超球判定法-1(モンテカルロ法)でも、超球判定法-2(球面四分木法)の場合でも同じである。このため、例えば超球判定法-1では、領域データによる簡略計算の適用によるランダム点探索の省略などが行われる。しかし、最大N点のランダム点を発生し、最悪の場合はN点についての相対位置計算が必要となるので、通常の相対位置計算のN倍の計算時間がかかることになる。拡大処理がネストになる場合、例えば、フィレット図形の相対位置手順では、さらにこの影響が大きくなる。そこで、逐次処理によっていた前述の超球判定法を、並列処理へ展開する高速化の手段を設けることによって、この問題の解決を図るものとする。

【0274】図43に、拡大手順の逐次構造と並列構造の概念を説明する模式図を示す。同図(a)は、図15の拡大手順s500において、s50435の超球判定手順(二次元でも三次元でもよい)を本体部313として、模式的に表現した逐次構造である。ここで、拡大手順30は図15の全体であり、超球判定法31は図のs50435の部分の意味する。拡大処理手順前段301は、図15のs50435より上に位置する部分であり、拡大処理手順後段302は、s50435より下に位置する部分である。

【0275】また、超球判定法31が超球判定法-1(図4.0)の場合、超球判定法の本体部313はt202超球判定法-11を指し、超球判定法前段311はt201を、また超球判定法後段315はt203を指す。同様に、超球判定法31が超球判定法-2(図4.2)の場合には、313はt303超球判定法-22を指し、311はt301~t302を、また315はt304を指す。

【0276】図43(b)に、同図(a)に対応する並

46

列構造を示す。超球判定本体部313を、一定の並列処理機構の基で連動する複数台(m台)の処理装置よりなり、各々の処理装置が分散手段(DISPATCH)312から別々のメッセージを受けてm分割された負荷を処理し、結果を連結手段(MERGE)314で連結することによって、最大でm倍の高速化を達成する機能を備えるものである。なお、このような並列計算処理機構の一般的な構造は、例えば、共立出版株式会社出版の「bit別冊;第5世代コンピュータの並列処理(瀧和男/編)」に記載のものが知られている。以下、具体的な並列化について詳しく説明する。

【0277】図44は、超球判定法-1の逐次処理を並列処理に展開した超球判定法-1Pを示すフローチャートである。

【0278】超球判定法-1P(t400)において、t401~t402は超球判定法前段であり、t401でメッセージ(空間点P、図形名G、半径r)を受信し、t402で所定の初期設定を行なう。t403(DISPATCH)では、まずt4031で空き状態にある処理装置{CPUi:i=1,m}を確保する。t4032で1CPU当たりの負荷として全試行回数Nをm等分した、試行回数n(=N/m)とする。次に、t4033とt40331では、前記m台のもとで超球判定法の本体部を並列処理させるべく、それぞれのCPUiの超球探索法-11Pにメッセージ(P、G、r、及び試行回数n)を送信する。

【0279】ここで超球探索法-11Pは本質的に超球探索法-11そのものである。t404(MERGE)では以下の処理を行なう。即ち、前記いずれかのCPUjより、相対位置データ(10j, dj, nil, 2)を受信する(t4041)。t4042で点Pの内外属性10jを判定し、もし、INであるならば、他の処理装置の結果を待たなくとも、この時点で交差が判明したので、t40421~t40423で所定のデータをセットし、DISPATCHされた全てのプログラムを停止させ、これらのCPU資源を開放した後、MERGEを抜ける。一方、OUTのときはt40424でdPを更新した後、t4043でDISPATCHされた全てのプログラムの返信が終了したかどうかを判定する。返信が終了していたら、MERGEを終わるものとし、そうでなければ、再び次の返信を待つ。t405は、超球判定法後段であり、求められた相対位置データ(10P, dP, nil, 2)を返信して終了する。

【0280】図45は、超球判定法-22(t303)の逐次処理を並列処理に展開した超球判定法-22Pを示すフローチャートである。超球判定法-22P(t503)において、t5031~t5032は、図42のt3031~t3037の内容を略記したものである。また、t5033は、t3038に対応する。ここで、1CPU当りの負荷を、四分割された区分矩形距離の一

一つ一つとする。

【0281】 $t5034$ (DISPATCH) では、まず $t50341$ で空き状態にある4台の処理装置 {CPU $i: i=1, 4$ } を確保する。このとき、自分自身も1台と数え、それでも不足する場合には重複して自分に割り当てる。次に、 $t50342$ と $t503421$ では、前記4台のもとで超球判定法の本体部を再帰的に並列処理させるべく、それぞれのCPU i の超球探索法-22Pにメッセージ: (P, G, r) と区分領域の中心 ($\theta 0, z 0$) 及びサイズ ($W/2, H/2$) を送信する。 $t5035$ (MERGE) の処理は、前述した $t404$ の処理と全く同じになるので説明を省略する。 $t5036$ では、求められた相対位置データ (IOP, d, P, nil, 2) を呼び出し側に返信して終了する。

【0282】以上のような並列処理構造をもつ図形処理装置の構成と動作の概略を説明する。並列図形処理装置の場合も、図2の構成が基本になる。但し、並列処理機構の基で並列的に動作する複数台 (m台) の処理装置 {CPU $i: i=1, m$ } に於て、主タスクが動作する処理装置 (例えば、CPU1) には、図2の11~15の機能が存在しており、他のCPU {CPU $i: i=2, m$ } には、少なくとも機能11、13と同等の機能を有して、超球判定法のサブタスクを並列処理する。また、記憶装置2の内容は、全てのCPUに共通であっても良いし、通信を介して互いに利用可能な、個々のCPU固有の装置に分散されて存在しても良い。但し、相対位置データ生成手順23には、並列化された超球判定法本体部313も含めるものとする。さらに、通信制御手段11には、異なるCPU間での通信機能も含まれる。

【0283】並列図形処理装置の動作は、図1の処理フローが基本になる。 $s101 \sim s103$ のメッセージは、並列計算処理機構の基では、同一CPU内だけでなく異なるCPU間でも行なわれる。主タスクが動作中のCPU1では、図1のメッセージ処理が全て行なわれる。他のCPUは、例えば、記憶装置2の相対位置生成手順23に超球判定法-11Pが実装されている場合には、CPU1の主タスク超球判定法-1Pからのメッセージ $s120$ を受けて、超球探索法-11Pをサブタスクとして、図40の手順 $t200$ に従って並列処理し、結果をCPU1の主タスクに返信する。

【0284】以上説明したように、拡大図形生成処理における超球判定処理を並列化することによって、本発明の相対位置データによる画像生成の一層の高速化が可能になり、三次元の複雑な図形処理においても十分に、実用に耐えるものとして行うことができた。なお、超球判定処理の並列処理は一例に過ぎず、繰返しや分岐等を伴う他の処理手順にも、同様にして適用できることは言うまでもない。

【0285】次に、三次元モデルによる応用図形処理の典型的な実施例として、図形表示 (走査線による表示と

パターン表示) について説明する。これらは、二次元モデルの場合の図形表示の自然な延長としてえられる。

【0286】図46に、境界面の走査線探索による図形表示の概念図を、図47に、その処理フローを示す。

【0287】三次元表示の手順 $t600$ は、二次元の表示手順 $s300$ (図8) を三次元に拡張したものである。即ち、対象図形Gの境界面を表示するために、視線方向に垂直な平面 π を視点の側に設定して、平面 π と直交する格子状の二次元走査線列を発生させ、この走査線列と図形Gの交点よりなる格子状の二次元境界点列を、相対位置データを繰返し計算することによって求め、この境界点列を曲面として表示することの特徴とする。

【0288】走査線による表示手順 $t600$ は、まず、 $t601$ で (視線方向ベクトル view, 図形名G) 及び、その他のパラメータ (走査開始点PP、走査線間隔W、走査線数 (N_i, N_j)、最大探索深さL、交差感度 ϵ) を受信する。 $t602$ で、viewに直交する平面上にあり、且つ互いに直交する単位ベクトル w_i, w_j を求める。 $t603$ で w_i 方向の外側ループを、その内側で、 w_j 方向のループを制御し ($t6032$)、 $N_i * N_j$ 本の走査線に関する境界点を求める。即ち、 $t6031$ で ($i, j=0$) の走査開始点 $P=P_{i0}$ を決め、 $t60321 \sim t60322$ で点Pの図形Gに関する相対位置データを求める。点Pが図形Gの内部にあるか外部にあるかによって、探索の向き v を決める ($t60323$)。

【0289】次に、 $t60324 \sim t60325$ のループ及び、 $t603251 \sim t603254$ は、二次元の表示手順 $s300$ の $s3044 \sim s3045$ 及び、 $s30450 \sim s30453$ と全く同じであり、ここで、境界点が存在する区間が決まる。この区間が存在する場合には ($t60326$)、 $t606261$ で境界点Q0を前記区間に関する二分法で求め、次の走査線 ($i, j+1$) の探索開始点を決める。区間が存在しない場合には、 $t603262$ でQ0をnilとし、次の開始点を P_{i0} より求める。 $t60327$ で、求められた境界点Q0を二次元アレーQQ (i, j) に格納する。最後に、 $t604$ で二次元アレーQQの全内容を曲面として表示する。この表示の特徴は、視線方向に関して見える部分だけが表示されることである。

【0290】図48~図50を用いて、三次元モデルにおけるパターン表示の実施例を説明する。二次元モデルでは、図20~23に示したように、四分木法を用いて空間を正方領域 (正方形) で細分することによって、パターン表示手順 $S700$ (図24) により表示結果を得るものであるが、三次元の場合には、八分木法を用いて空間を立方体で細分することによって、表示結果を得る。

【0291】このために、あらかじめ、対象とする図形Gを包む立方体を設定し、この立方体を基に、領域感度

を与えられた八分木法により、図形Gを正方領域（立方体セル）で細分し、その各正方領域の中心を空間点とする相対位置データの内外判定データから、図形領域に対する各正方領域の内、外、未定の領域判定を行ない、未定の領域について上記一連の処理を所定領域感度まで繰り返して後、残っている各未定の領域（未定セル）の各面に対して、その四頂点について得た内外判定データの組み合わせパターンに応じて、あらかじめ定めた所定の境界辺セグメントを求め、各面の境界辺セグメントを所定の手順の基で結合してえられる表示面セグメントを、当該未定領域の表示面とするものである。

【0292】パターンを決めるための未定セルを得るまでの手順は、図20～21におけると同様であるので説明を省略し、その後の、未定セルに対する手順について、図48を用いて説明する。図48(a)は、未定セルの一つを表している。これら一つ一つの未定セルの各面に対して、例えば、同図(b)のようにベクトル表記の境界辺セグメントL11～L15、及び、L21～L23を求める。この手順は、図22～23に於けるものと同じである。ベクトル表記の仕方は、例えば、INの頂点を右に、OUTの点を左に見るように向きを与えるものとする。

【0293】次に、これらの境界辺セグメントを、隣接するもの同士をまとめてグループ化すると、例えば、同図(c)のように、向き付けられた二つの多边形、L11～L15及び、L21～L23が得られる。さらに、これらの多边形をそれぞれ、その頂点の重心に関して三角形分割する事により、表示面セグメントを求める。即ち、同図(d)に示すようにC1に関して三角形分割して得られる面F1と、三辺形のため分割されない三角形領域F2が表示面セグメントとなる。なお、面の表側は、多边形に関して右ネジの向きである。これらの表示面セグメントを全ての未定セルについて集めると、この集合全体は、図形Gの境界面の（与えられた領域感度の精度を持つ）三角形近似となっている。このデータを基に、任意の視点からの表示を行なうことは容易である。

【0294】図49は、表示結果の一例である。表示した形状Gは概略、以下のように記述される。

【0295】RE1 = (四角柱；中心O1、三辺の長さd1, w1, h1)

RE2 = (四角柱；中心O2、三辺の長さd1, w1, h1)

RE3 = (四角柱；中心O3、三辺の長さd2, w2, h2)

CY = (円柱；軸方向の中心O4、半径R1, 高さh3)

G1 = (+; ; RE1, RE2)

G2 = (-; ; G1, RE3)

G3 = (-; ; G2, CY)

すなわち、同等の四角柱RE1とRE2の和図形G1か

ら、四角柱RE3の差図形、さらに円柱CYの差図形による立体図形Gである。

【0296】図49(a)は、図20～図21と同様に求めた未定セルをそのままセル（直方体）表示したものである。一方、同図(b)は、これらの未定セルから、本手法によりパターン表示を行なった結果を示している。両者を比較してわかるように、本実施例のパターン表示によれば、同一領域感度の場合、詳細な形状表面の表示が可能になる。

【0297】図50は、三次元モデルにおけるパターン表示の手順を示すフローチャートである。パターン表示(t700)は、まず、t701でメッセージ（正方領域の中心O、幅W、領域感度δ、図形名G）を受信し、t702で開始点Pとセル幅wを初期化し、メッセージ(P, w, G)を送って、再帰的に八分木法を適用し、図形Gの境界面の三角形近似を行なうものである(t703)。

【0298】即ち、t7031～t7032で点Pの図形Gに関する相対位置データ(IO, d, Q, KIN D)を求め、t7032で、点Pを中心として半径dの球が、点Pに関する幅wのセルを含む場合には（同図(b)）、このセルは内セル、または外セルのいずれかであるので、なにもしないで呼び出し側に戻る。t7034でwが領域感度δ以下である場合には、これ以上分解されない未定セル、即ち、境界セルとなるのでt70341～t70435で前述した手順にしたがって、表示面セグメントを作成する。セル幅wが領域感度δより大きい場合には、 $w' = w/2$ とし(t7035)、t7036で中心Pを通り且つ、x、y、z軸に直交する三つの平面でセルを8等分し、それぞれの中心Pi (i = 1, 8)を求める。

【0299】次に、これら8箇のサブセルに対して、t7037、t70371で順次、メッセージ(Pi, w', G)を送って、八分木法で再帰的に評価する。このフローでは、表示面セグメントから実際の表示を行なう部分は自明なものとして省略した。

【0300】本実施例によれば、表示面セグメントの全集合が図形Gの境界面全体の三角形近似である表示面データとなるので、任意の視点に対する表示に適用でき、また、詳細な形状表面の表示が可能になる。

【0301】次に、三次元モデルによる応用図形処理の他の実施例として、体積計算について説明する。体積計算は、前述の二次元における面積計算(s800)の単純な拡張として得られる。

【0302】即ち、あらかじめ、対象とする図形Gを含む立方体を設定し、この立方体を基に、領域感度を与えられた八分木法により、図形Gを正方領域（立方体セル）で細分し、その各正方領域の中心を空間点とする相対位置データの内外判定データから、図形領域に対する各正方領域の内、外、未定の領域判定を行ない、未定の

領域について上記一連の処理を所定領域感度まで繰り返して後、上記の処理過程で前記内部セルとして判定された全てのセルの体積の総和を求めるものである。

【0303】図51に、体積計算の処理フローを示す。図示のように、図27の面積計算フローにおける細分過程が四分木法から八分木法に置換されたのみであり、説明を簡単に留める。t8033では、点Pを中心として半径dの球が、点Pに関する幅wのセルを含む場合（同図(b)）、t80331で内セルと判定された場合には、t803311でセル体積の積算を行なう。または、t8033でセル幅wが領域感度δ以下の場合、セルが境界セルとなるので、その中心が図形内部にある場合のみ（t80331）、体積の積算を行なう（t803311）。

【0304】次に、三次元モデルにおける他の応用処理として、フィレット図形処理の実施例を図52～図55にしたがって説明する。フィレット図形処理とは、与えられた図形G（図52(a)）の凸部を半径r1、凹部を半径r2で丸めた図形G*（図53(f)）を生成するもので、三次元のオフセットや反転を組み合わせることで処理される。

【0305】図54に、図形G*の生成手順のトリートメント、図55に、対応する図形オブジェクトのデータ構造を示す。なお、ここでのオフセットは、前述のように拡大図形における操作図形に球を用いている。

【0306】応用図形処理機能15は、メッセージ【フィレット；パラメータ（r1, r2）；対象図形名G】を受信すると、フィレット図形G*を以下の手順で生成する。

【0307】G1 = (オフセット；r2；G)
G2 = (反転；G1)
G3 = (オフセット；(r1+r2)；G2)
G4 = (反転；G3)
G* = (オフセット；r1；G4)

ここで、G1～G4は図52(b)～図53(e)に対応している。

【0308】次に、三次元モデルによるさらに好適な応用図形処理の実施例として、NC工具経路計算への適用例を、図56～図63を用いて説明する。金型加工などにおける輪郭切削では、NC工作機械の工具の切削経路を表す工具経路データ（NC情報）を求める必要がある。以下では、三軸のNCフライス加工を対象として説明する。

G* = (ボールエンド拡大；半径r；図形G) … (様式14)

ここで、G*の生成手順である「ボールエンド拡大」は、前述のように操作図形として球を用いる三次元モデルに拡張された「三次元の拡大」と同じであり、あらためて用意する必要はない。別名を付けて図形生成の相対位置生成手順ファイル23に登録しておけばよい。但し、図形Gは、例えば図38(a)のようにあらかじめ

* 【0309】図56(a)に、輪郭切削に用いられる(a1)ボールエンドミル、(a2)フィレットエンドミルおよび(a3)フラットエンドミルの三種類の工具の断面図、同図(b)に各々の先端部(歯面)の模式図を示す。図示の点Oを工具中心とし、各歯面の形状である球体、扁平球、円板に、オフセットの操作図形が対応される。

【0310】図57に、切削加工の対象となる金型形状Gの側面図を示す。加工は金型上面F1とF2に対して行われ、工具を垂直に保って歯面をF1やF2に接しながら切削するものとする。このときの工具経路データとは、工具が上面F1、F2をスキャンするための工具中心Oの軌跡である。

【0311】図58は、従来方法の工具経路を示す説明図である。図示のように、F1、F2の面毎に、面の各点(Q1/Q2)から外向きの法線ベクトル(v1/v2)を出して、その点から半径rの点(P1/P2)を工具経路の候補点とする。このようにして、工具経路CL1ないしCL2面を求める。

【0312】この結果、同図(a)のように、形状Gの面F1、F2が凹に交差する場合は、CL1とCL2の相貫部に関係して除去されるべき部分を、CL1とCL2の交線Lを求めてカットする。これによって図の点P'は工具経路に含まれない。一方、同図(b)のように、形状Gの面F1、F2が凸に交わる場合には、法線ベクトルから計算したCL1とCL2は相貫部付近でスプリットしてしまうので、この場合には、両者を適当な条件の基で延長して、交線Lを求めなければならない。一般に、これらの処理は煩雑であり、三個以上の面が同一箇所でも相貫する場合などでは、この傾向がより顕著になる。

【0313】図59に、本実施例による工具種別毎の工具経路の模式図を示す。ボールエンドミルによる工具経路は、同図(a)のCL1およびCL2ようになる。この工具経路データは、三次元モデルによる図形の拡大手順（ここでは、オフセットと同義）を金型図形Gに対して作用させることによってえられる。

【0314】即ち、図形Gの各点Pに対応させる操作図形g(P)として、ボールエンドミル先端部に対応する半径rの球を設定する（図52(b1)）。次に、様式14によるメッセージを応用図形処理機能15に送り、拡大図形G*を生成する。

* 【0315】

生成されている必要がある。

【0316】図60に、ボールエンドミルによる工具経路計算の概略処理フローを示す。工具経路データ生成手順t1000は、空間点P、図形名Gと共にボールエンドミル先端部(歯面)に対応する半径rを受信する(t1001)。次に、金型図形Gを半径r分オフセットし

た形状 G^* を求める(t1002)。この形状 G^* の境界面が工具の中心の移動可能な場所である。次に、工具軸方向に垂直な平面上で工具軸に垂直な方向の移動点列 $\{P_i\}$ を設定する(t1003)。そして、点列の数だけt1004のループ処理を行う。即ち、t10041で、点 P_i から工具軸方向に一致するプローブ(ここでは、境界探索用の走査線)を出す。t10042で、プローブに沿って図形 G^* の境界点 Q_i を求める。この点列 $\{Q_i\}$ が求める工具経路データとなる。なお、このような境界点の探索は、図8で説明した図形表示の走査線探索と同じ処理である。

【0317】ところで、金型形状 G に関する「三次元の拡大」の手順で得られる図形 G^* は、図形 G の上面 F_1 、 F_2 から求めたい工具経路 CL_1 および CL_2 に一致することを、補足的に説明する。

【0318】まず、次の事実に着目する。(1) 工具中心 P が金型形状 G の内部にあれば、工具は金型と交差し点 P は工具径路面の内側にある。(2) 工具中心 P が金型形状 G の外部にあって、その境界との距離が工具半径 r より大ならば、工具中心 P は工具径路面の外側にある。(3) 工具中心 P が金型形状 G の外部にあって、その境界との距離が工具半径 r より小ならば、工具中心 P は工具径路面の内側にある。(4) 工具中心 P が工具経路面上にあるのは、点 P が形状 G の外部にあって境界との距離が r に等しいときである。

【0319】従って、本実施例の「ボールエンド拡大」、即ち「三次元の拡大」が決定し、空間点 P に対する図形 G^* の相対位置データは、点 P を工具中心とし、図形 G^* の境界を工具径路面と見做したときの解釈に全く一致する。このことは、図形 G が、第一種である場合には、明らかである。

【0320】因みに、図58(a)における点 P' は、距離として、 $P'Q'$ ($=r$)でなく、 $P'Q''$ ($<r$)が与えられるので、図形 G^* の内部、従って、工具径路面の内側にあると自動的に判定される。ところで、「三次元の拡大」の場合、図形 G が第二種である場合に*

$G^* = (\text{フィレットエンド拡大 ; 半径 } r ; \text{図形 } G) \quad \dots (\text{様式 } 15)$

として生成する。ここで、 G^* の生成手順である「フィレットエンド拡大」は、あらかじめ相対位置生成手順ファイル23に作成しておく。

【0325】しかし、フィレットエンドミル先端部の偏平球は、工具中心 O を中心とする球の変形としての対称図形であるので、拡大の手順を一部変更する必要がある。この場合の図形 G^* の生成手順の概略は以下のようになる。

【0326】(1) 空間点 P に対する図形 G の相対位置データを求めて、($IO, d, Q, KIND$)とする。

(2) $IO = IN$ のときは、点 P の G^* に置ける相対位置データを、($IO, d+h, Nil, 2$)とする。

(3) $IO = OUT$ で、 $d \geq r$ のときは、相対位置デー

*は、距離データが下界の近似値として与えられるので、上記の事実だけでは正確な内外判定が出来ない。とくに、工具中心 P が図形 G の外部にあり、図形 G との距離が、 r より小さい近似値として与えられる場合には別の判定法が必要である。この場合、点 P を中心とするボールエンドミルの先端部の球(図56(b1))が図形 G と交差するかどうかを判定することによって、点 P の内外関係を決定することができる。本実施例の「三次元の拡大」に用いられる前述の超球判定法-1(または2)は、真に、この操作を具体化した手順であり、以下では特に「超球拡大」と呼ぶことにする。

【0321】さらに、「超球拡大」は、図形 G の境界である F_1 および F_2 を直接扱わず、立体である図形 G そのものを拡大するので、図58(b)のような図形のスプリットは起こりえず、必ず閉じた図形として図59のように生成される。結局、工具経路のデータ、即ち、工具軸方向のプローブによる探索によって求められる図形 G との距離が r となる点列は、 CL_1/CL_2 相貫部に関する特別の例外処理を不要としている。

【0322】図形 G^* は、 CL_1 、 CL_2 を境界の一部として含む図形 G の拡大図形となる。但し、図形 G^* が生成されたと言っても、この段階では、図形 G^* が(様式14)の形で登録しただけで、 CL_1 、 CL_2 を陽に与えるものではない。そこで、前述の表示手順t600により、図61に示すように、工具軸方向に一致する垂直な視線方向のプローブを出して、図形 G^* の境界点を探索し、 CL_1 、 CL_2 を点列 $\{Q_i\}$ として求める。

【0323】次に、フィレットエンドミルによる工具経路は、図59(b)の CL_1 および CL_2 となる。この工具経路データも、基本的には「三次元の拡大」を金型図形 G に対して適用することによってえられる。

【0324】即ち、図形 G の各点 P に対応させる操作図形 $g(P)$ として、フィレットエンドミル先端部(図56(b2))に対応するパラメータ(r, h)の偏平球を設定する。次に、この拡大図形 G^* を、

タを($IO, d-r, Nil, 2$)とする。(4) $IO = OUT$ で、 $d < r$ のときは、超球判定法のアルゴリズムを球に代えて偏平球に適用して、相対位置データを求める。

【0327】ここで、上記(2)と(3)の手順が可能となる理由は、偏平球が点 P を中心とする半径 r の外接球と半径 h の内接球を持つことから、点 P が図形 G の内部にあるときは内接球で、また、点 P が図形 G の外部にあって、図形 G^* の境界から十分に離れた点 P では、半径 r の外接球で前記偏平球を近似できるからである。なお、偏平球に関する超球判定法については、モンテカルロ法及び四分木法とも可能であるが、それぞれ、図40及び図42の自然な変形として容易に構成できるもので

ある。

【0328】このようにして生成された図形G*の境界上面F1、F2が工具経路面CL1、CL2と一致することは、図62に示すように、偏平球が点対称の図形であるということから明らかである。即ち、点Q*を図形G*の境界上の点とするならば、点Q*に対して図形Gの境界点Qが存在して、偏平球g(Q)は点Q*で、偏平球g(Q*)は点Qで、それぞれ図形G*およびGの境界に接する。CL1、CL2の工具経路データを点列として求める手順は、前述のボールエンドミルの場合と*10

$$G^* = (\text{フラットエンド拡大; 半径 } r; \text{図形 } G) \quad \dots (\text{様式 } 16)$$

として生成する。ここで、G*の生成手順である「フラットエンド拡大」は、あらかじめ相対位置生成手順ファイル23に作成しておく。

【0331】ところで、半径rの円板は、フィレット半径hをゼロとしたときのパラメータ(r, 0)の偏平球と解釈できるので、「フラットエンド拡大」の手順は、「フィレットエンド拡大」の特別の場合として扱うことができる。従って、以降の説明は、フィレットエンドミルの場合と同じであり省略する。

【0332】以上、3種類の工具による工具経路計算について説明した。本実施例によれば、加工面の関連部における従来のような後処理が不要になる。

【0333】ところで、NCのオペレーションプランニングにおいては、シミュレーションによって加工手順の妥当性を検証するが、その際、工具経路データに基づい※

$$\begin{aligned} G1 &= (\text{ボールエンドミル; 半径 } r; \text{図形 } G) \\ &= (\text{反転}; ; (\text{ボールエンド拡大; 半径 } r; (\text{反転}; ; \\ &\quad (\text{ボールエンド拡大; 半径 } r; \text{図形 } G))) \quad \dots (\text{様式 } 17) \end{aligned}$$

生成された図形G1は、工具が加工後に残す図形：(図30★はじめ相対位置生成手順ファイル23に作成しておくべき図形G+削り残し領域Gr)を表す図形となっている。図形G1を生成する手順「ボールエンドミル」は、あらかじめ★

$$\begin{aligned} Gr &= (-; ; \text{図形 } G1, \text{図形 } G) \\ &= (\text{反転}; ; (\text{和}; ; (\text{反転}; ; G1), \text{図形 } G)) \quad \dots (\text{様式 } 18) \end{aligned}$$

として、「図形G1-図形G」を行なえば、削り残し領域が生成される。

【0337】削り残し領域図形Grの生成は、Gr相対位置データ生成手順を登録しただけであるので、さらに、該領域Grを表示したり、その大きさ(体積)を測る場合には、前述した図形表示または体積計算に関する☆40

$$\begin{aligned} G1 &= (\text{フィレットエンドミル; 半径 } r, \text{半径 } h; \text{図形 } G) \\ &= (\text{反転}; ; (\text{フィレットエンド拡大; } r, h; (\text{反転}; ; \\ &\quad (\text{フィレットエンド拡大; } r, h; \text{図形 } G))) \quad \dots (\text{様式 } 19) \end{aligned}$$

その後の処理は、ボールエンドミルの場合と同じである。

【0340】次に、フラットエンドミルによる削り残し◆

$$\begin{aligned} G1 &= (\text{フラットエンドミル; 半径 } r; \text{図形 } G) \\ &= (\text{反転}; ; (\text{フラットエンド拡大; 半径 } r; (\text{反転}; ; \\ &\quad (\text{フラットエンド拡大; 半径 } r; \text{図形 } G))) \quad \dots (\text{様式 } 20) \end{aligned}$$

以上、加工工程のオペレーションプランニングに、本実

*同じである。

【0329】さらに、フラットエンドミルによる工具経路は、図59(c)のCL1およびCL2となる。この工具経路データも、基本的には「三次元の拡大」を金型図形Gに対して適用することによってえられる。

【0330】即ち、図形Gの各点Pに対応させる操作図形g(P)として、フラットエンドミル先端部(図56(b3))に対応する半径rの円板を設定する。次に、この拡大図形G*を、

※で行なう、ワーク削り残し部の検出は重要なチェック項目の一つである。前述の状況のもとで、三軸NCフライス加工を仮定すると、F1とF2の相貫部において、図63のような削り残し領域が発生する。本発明の相対位置データによる図形処理は、三次元モデルによるNC削り残し領域の検出にも有効に適用できる。

【0334】NC削り残し領域の検出は、前述の3種類の工具を例に説明すると、以下の手順によって処理される。

【0335】ボールエンドミルによる削り残し領域は、まず、金型図形Gとボールエンドミル先端部(歯面)に対応する半径rから、様式17による図形G1を生成する。

【0336】

30★はじめ相対位置生成手順ファイル23に作成しておくべきもので、前述の「ボールエンド拡大」と「反転」手順の合成として作成される。そこで、

☆応用処理の手順を適用すればよい。

【0338】次に、フィレットエンドミルによる削り残し領域は、同様にして様式(19)による図形G1を生成する。

【0339】

◆領域は、同様にして様式(20)による図形G1を生成する。

【0341】

を適用することによって、従来は例外処理によっていたものを、工具経路計算からの一貫処理として統一的に解決することができる。

【0342】

【発明の効果】本発明の図形処理方式によれば、階層関係にある図形間の動的な通信制御方式が可能になるので、相対位置データを用いる図形処理を効率的に実行できる効果がある。

【0343】本発明の図形処理方式によれば、領域データを用いて相対位置データの計算を簡略化するので、処理を高速化できる効果がある。

【0344】本発明の図形処理方式によれば、拡大図形の処理を操作図形方式による場合分けを行うので、処理がシンプル化され且つ処理精度が向上できる効果がある。特に、従来は真の距離データで処理しなければならない場合にも、超球判定機能による内外判定を可能にしたことで、二次元のみならず三次元図形の拡大処理とそれを含むオペランド図形処理に、例外なく相対位置データの適用が可能となり、信頼性の向上と適用範囲の拡大を図り且つロバストで使い勝手のよい図形処理方法および装置を提供することができる。

【0345】本発明の図形処理方式によれば、相対位置データを用いる表示処理において、新規な視線方式および/または四分木パターン方式を可能にしたので、表示処理を実用化可能なレベルに高速化し且つ表示精度を向上できる効果がある。

【0346】本発明の図形処理方式によれば、超球判定など時間を要する繰返しまたは分岐処理部に並列処理構造を適用したので、複雑な図形処理の高速化を達成できる効果がある。

【図面の簡単な説明】

【図1】本発明の一実施例を示し、図形処理装置の全体的な図形処理方法の処理手順を示すフローチャート。

【図2】本発明の一実施例による図形処理装置の構成を示す機能ブロック図。

【図3】図形相対位置生成手順の格納形式及び図形オブジェクトデータの階層構造を示す説明図。

【図4】表示処理手順及び応用処理手順の格納形式を示す説明図。

【図5】円図形の相対位置生成手順の説明図及びフローチャート。

【図6】円図形の第一種の相対位置データによる表示手順の説明図及び相対位置データテーブル。

【図7】円図形の第二種の相対位置データによる表示手順の説明図及び相対位置データテーブル。

【図8】本実施例による視線方式表示処理手順を示すフローチャート。

【図9】視線方式表示処理の探索可視点の決定方法を示す説明図。

【図10】視線方式表示処理における交差感度の説明

図。

【図11】本実施例による和(+)の相対位置データ生成手順を示すフローチャート。

【図12】和(+)の場合分け処理の説明図。

【図13】和図形の表示手順の説明図及び相対位置データテーブル。

【図14】操作図形方式による拡大処理の説明図。

【図15】本実施例による操作図形方式の拡大処理手順を示すフローチャート。

【図16】超球判定方法の説明図。

【図17】本実施例による超球判定方式の内外判定手順を示すフローチャート。

【図18】拡大図形の表示手順の説明図及び相対位置データテーブル。

【図19】超球判定法を利用する楕円の拡大図形表示手順の説明図及び相対位置データテーブル。

【図20】四分木法による正方領域の分割処理を示す説明図。

【図21】正方領域の内外判定基準を示す説明図。

【図22】四分木法による表示処理を示す説明図。

【図23】未定セルの四頂点内外判定データパターンと表示セグメントを示す説明図。

【図24】本実施例による四分木式パターン型表示処理手順を説明するフローチャート及び説明図。

【図25】本実施例による通信制御機能の動作を示す説明図。

【図26】四分木法による面積計算の説明図。

【図27】本実施例による面積計算の処理手順を説明するフローチャート及び説明図。

【図28】本実施例によるフイレット処理の説明図。

【図29】静的領域データの概念を示す説明図。

【図30】動的領域データの概念を示す説明図。

【図31】領域データを利用する和図形生成のトリート構造とオブジェクトデータを示す説明図。

【図32】静的領域データの領域を示す説明図。

【図33】本実施例による静的領域データを利用する和図形の相対位置データ生成過程を示す説明図。

【図34】本実施例による動的領域データを利用する和図形の相対位置データ生成過程を示す説明図。

【図35】本実施例による領域データを利用する相対位置データ生成処理のフローチャート。

【図36】三次元モデルにおける基本図形群の説明図。

【図37】三次元モデルの一例を示す金型形状の外観図。

【図38】本実施例による金型形状(図37)の図形生成手順の概要を示す説明図。

【図39】本実施例による金型形状(図37)のオブジェクトデータの構成図。

【図40】本実施例によるモンテカルロ法による三次元超球判定法の処理を示すフローチャート。

59

【図41】球面四分木法による三次元超球判定法の説明図。

【図42】本実施例による球面四分木法による三次元超球判定法の処理を示すフローチャート。

【図43】本実施例による拡大手順の逐次処理と並列処理を対比的に示す模式図。

【図44】本実施例によるモンテカルロ法による三次元超球判定法の並列処理を示すフローチャート。

【図45】本実施例による球面四分木法による三次元超球判定法の並列処理を示すフローチャート。

【図46】走査線探索方式による三次元表示の説明図。

【図47】本実施例による走査線探索方式の三次元表示処理を示すフローチャート。

【図48】パターン方式による三次元表示の説明図。

【図49】本実施例によるパターン方式による三次元表示の説明図。

【図50】本実施例によるパターン方式による三次元表示処理を示すフローチャート。

【図51】本実施例による体積計算のフローチャート。

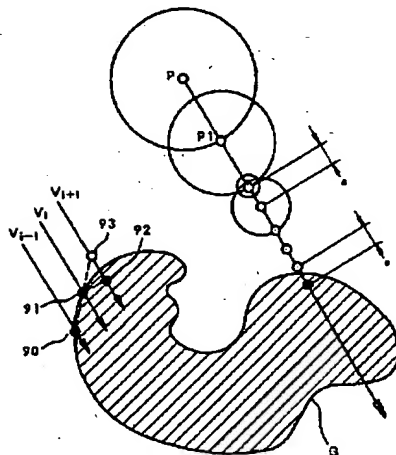
【図52】三次元モデルによるフィレット処理の説明図。

【図53】三次元モデルによるフィレット処理の説明図。

【図54】フィレット処理のトリー構造図。

【図9】

図 9



60

【図55】本実施例によるフィレット処理のオブジェクトデータの構成図。

【図56】NCフライス加工工具の形状図。

【図57】切削加工の対象となる金型形状の側面図。

【図58】従来の工具経路の説明図。

【図59】本実施例により得られる工具経路の模式図。

【図60】本実施例によるボールエンドミルの場合の工具経路データの生成手順を示すフローチャート。

【図61】工具軸方向プローブによる工具経路生成手順の説明図。

【図62】フィレットエンドミル先端部の扁平球の点対称の説明図。

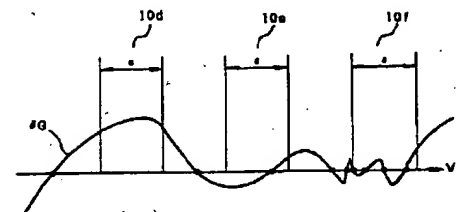
【図63】本実施例によるNC削り残し領域の検出処理を説明する模式図。

【符号の説明】

1…CPU、2…メモリ、3…入力装置、4…表示装置(CRT)、11…通信制御機能、12…図形オブジェクトデータ生成機能、13…図形相対位置データ生成機能、14…図形表示処理機能、15…応用図形処理機能、23…相対位置データ生成手順(プログラム)格納エリア、24…図形オブジェクトデータ格納エリア、25…表示制御手順格納エリア、26…応用図形処理手順格納エリア、IO…内外判定データ、d…距離データ、Q…近地点、KIND…種別フラグ。

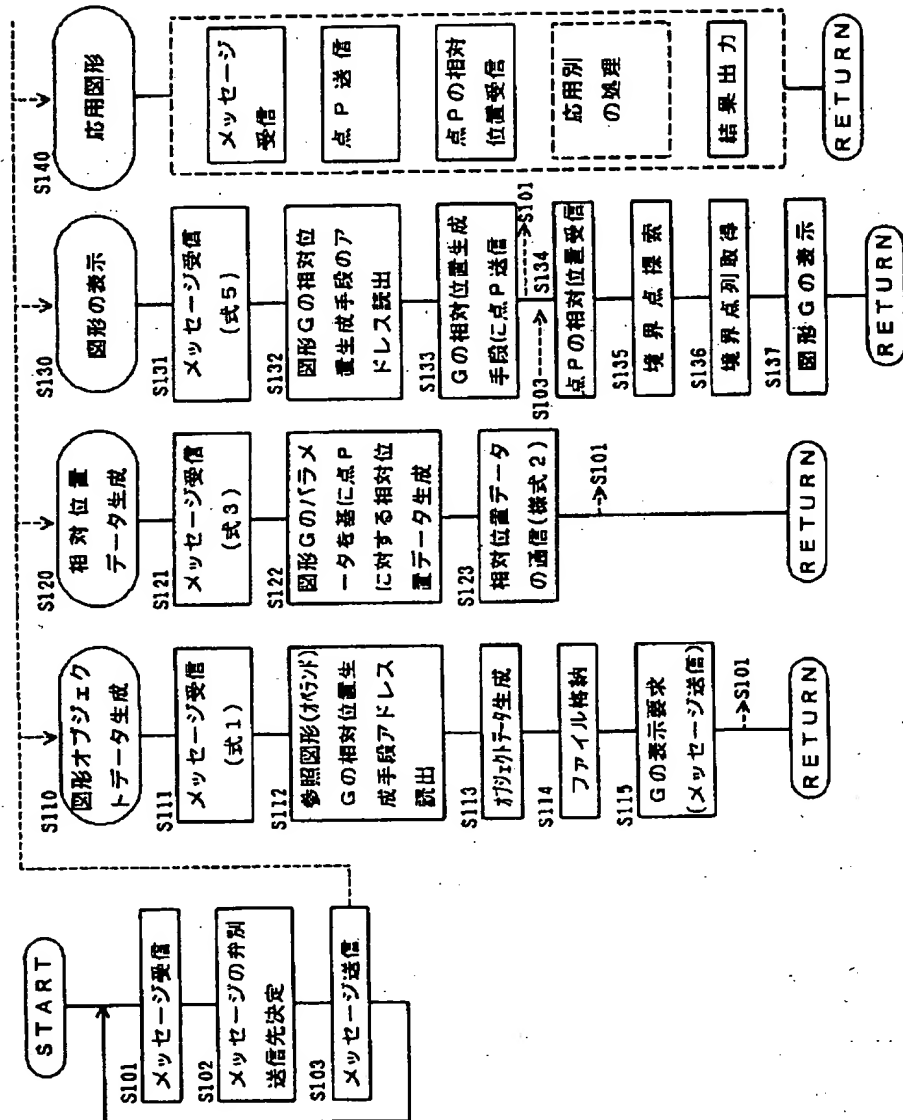
【図10】

図 10



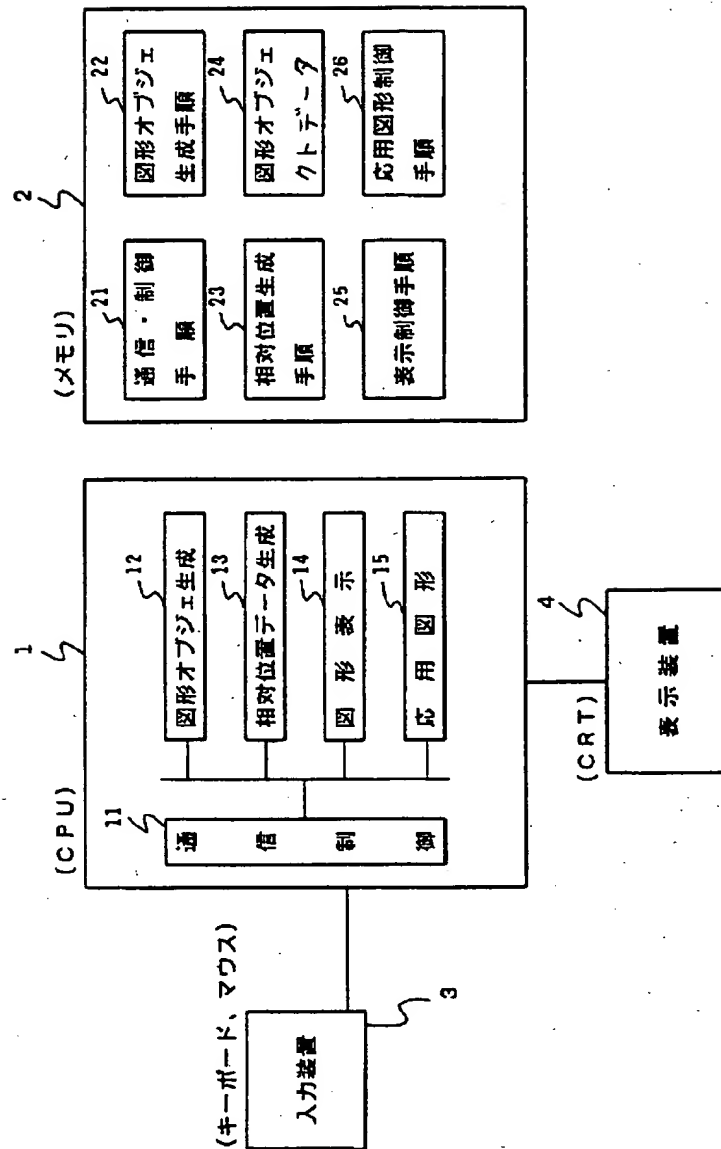
【図1】

図 1



【図2】

図 2



【図3】

図 3

(a)

#ccc	円図形の相対位置生成手順	231
#sss	四角形の "	232
#ttt	三角形の "	234
	∫	
#rrr	反転図形の相対位置生成手順	235
#aaa	和(+)図形の "	236
#zzz	拡大図形の "	237
	∫	
#yyy	超球判定処理手順	238

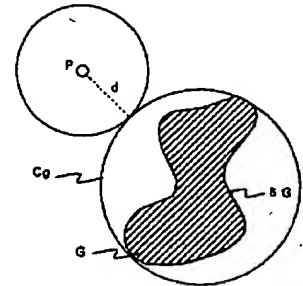
(b)

#obd	図形名 : D	
	手順ポインタ : #zzz	→ 拡大図形の相対位置生成手順
	図形ポインタ : #ob+	
	パラメータ(r):	
#ob+	図形名 : C	
	手順ポインタ : #aaa	→ 和(+)図形の相対位置生成手順
	図形ポインタ : #oba	
	図形ポインタ : #obb	
#oba	図形名 : A	
	手順ポインタ : #ccc	→ 円図形の相対位置生成手順
	パラメータ : 中心座標値と半径の値	
#obb	図形名 : B	
	手順ポインタ : #sss	→ 四角図形の相対位置生成手順
	パラメータ : 中心座標値と二辺の長さ	

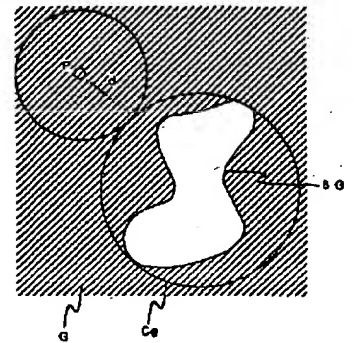
【図29】

図 29

(a1)

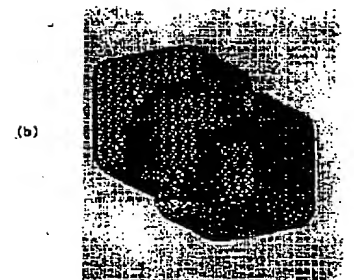
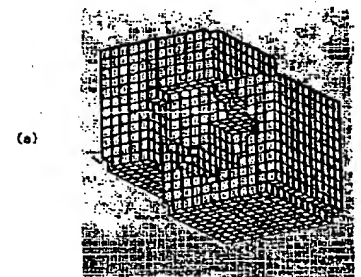


(a2)



【図49】

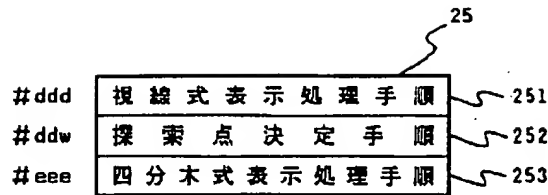
図 49



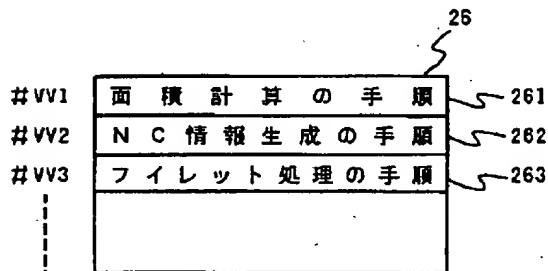
【図4】

図 4

(a)

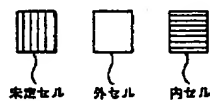
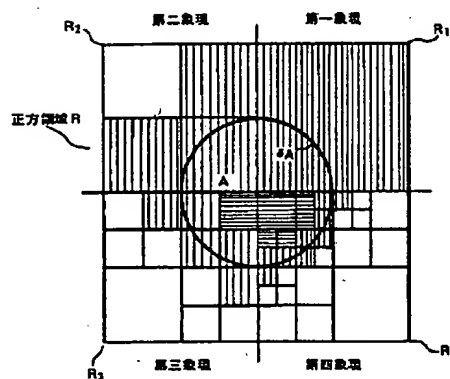


(b)



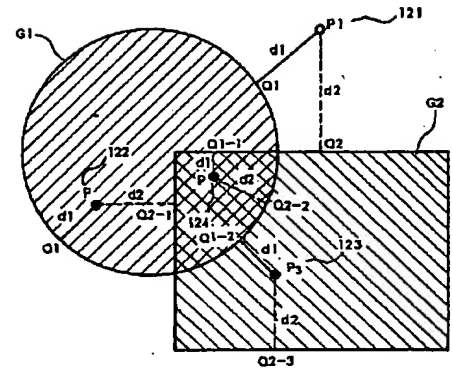
【図20】

図 20



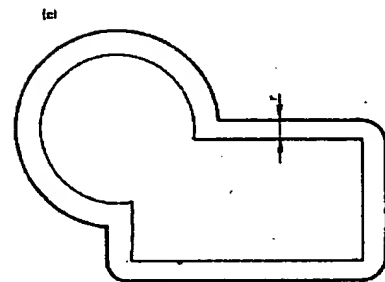
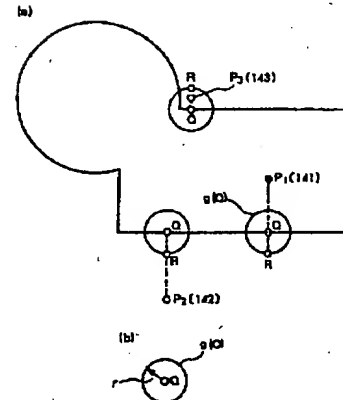
【図12】

図 12



【図14】

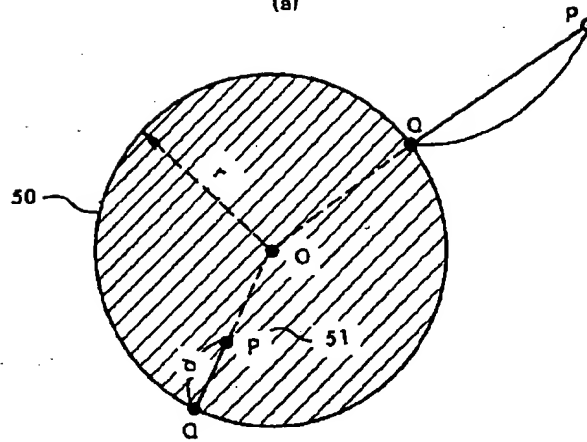
図 14



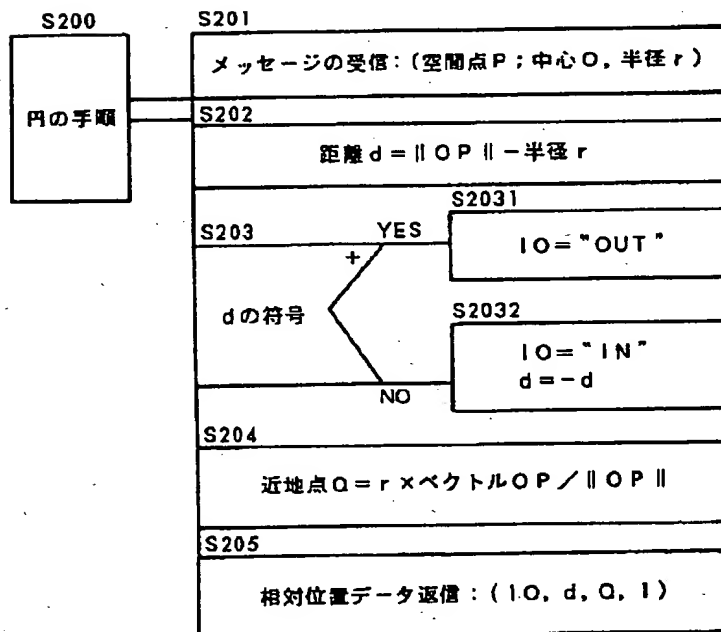
【図5】

図 5

(a)

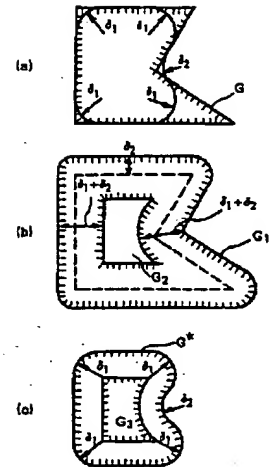


(b)

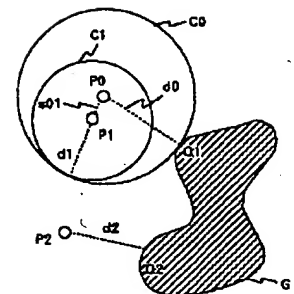


【図28】

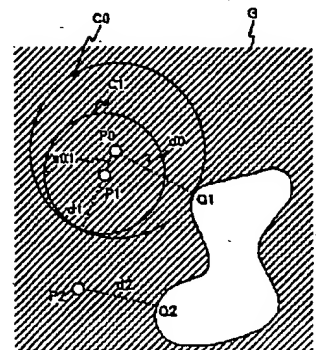
図 28



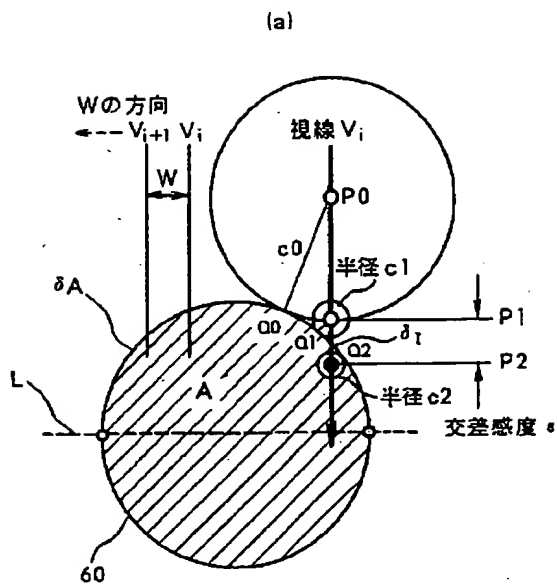
【図30】

図 30
(b1)

(b2)



【図6】

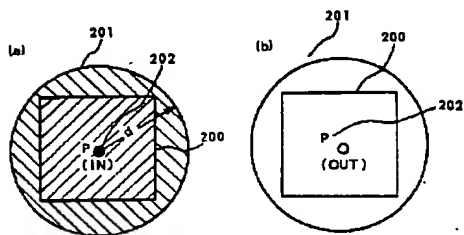
 6

(b)

メッセージ 探索点 P	相対位置データ r_data(A)	備 考
P0 (開始点)	(OUT, c0, Q0, 1)	
P1 (P0P1=c0)	(OUT, c1, Q1, 1)	c1 < 交差感度 e
P2 (P1P2=c)	(IN, c2, Q2, 1)	

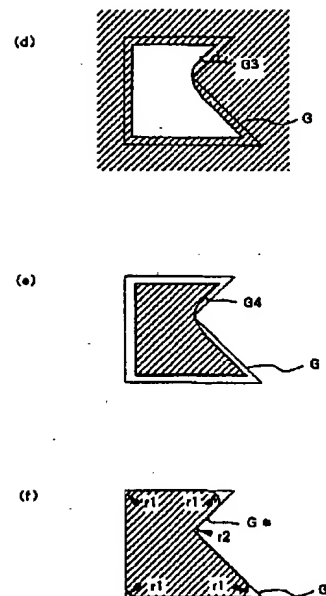
【圖 2 1】

21



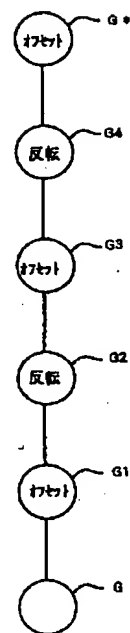
【图 5 3】

53



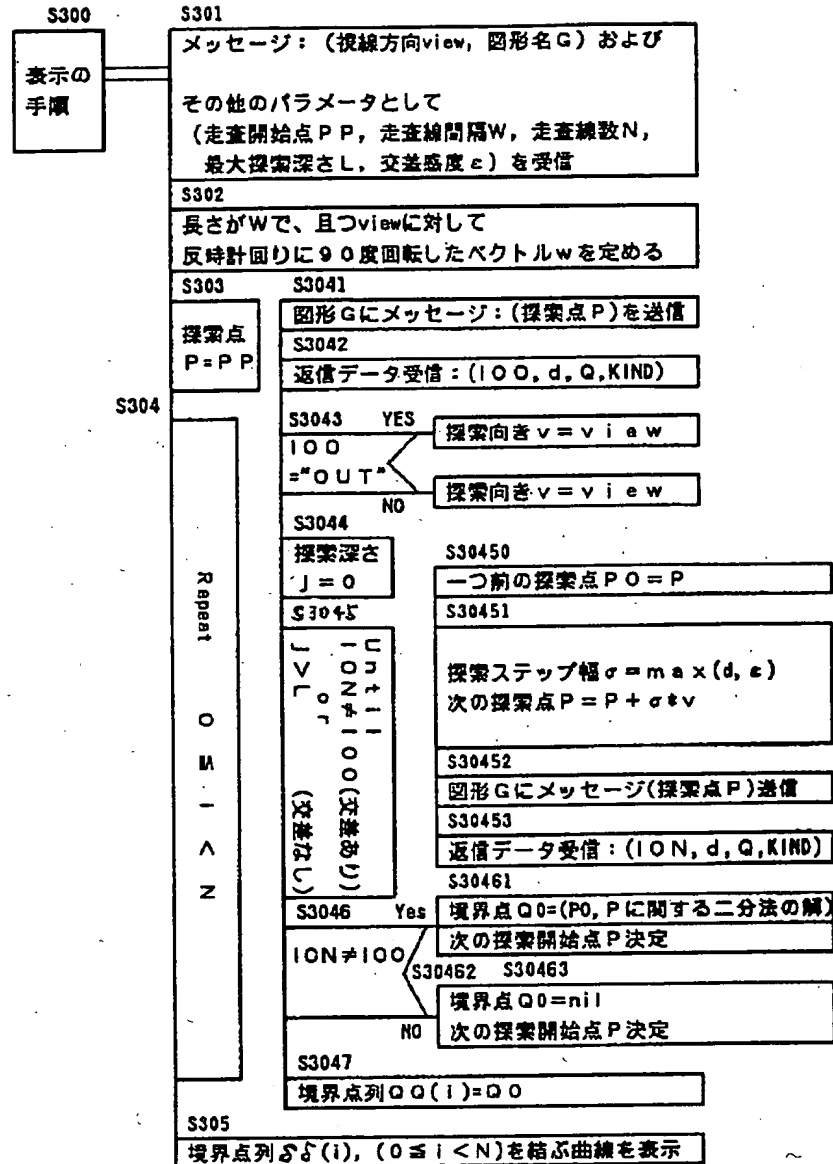
【图 5 4】

图 54



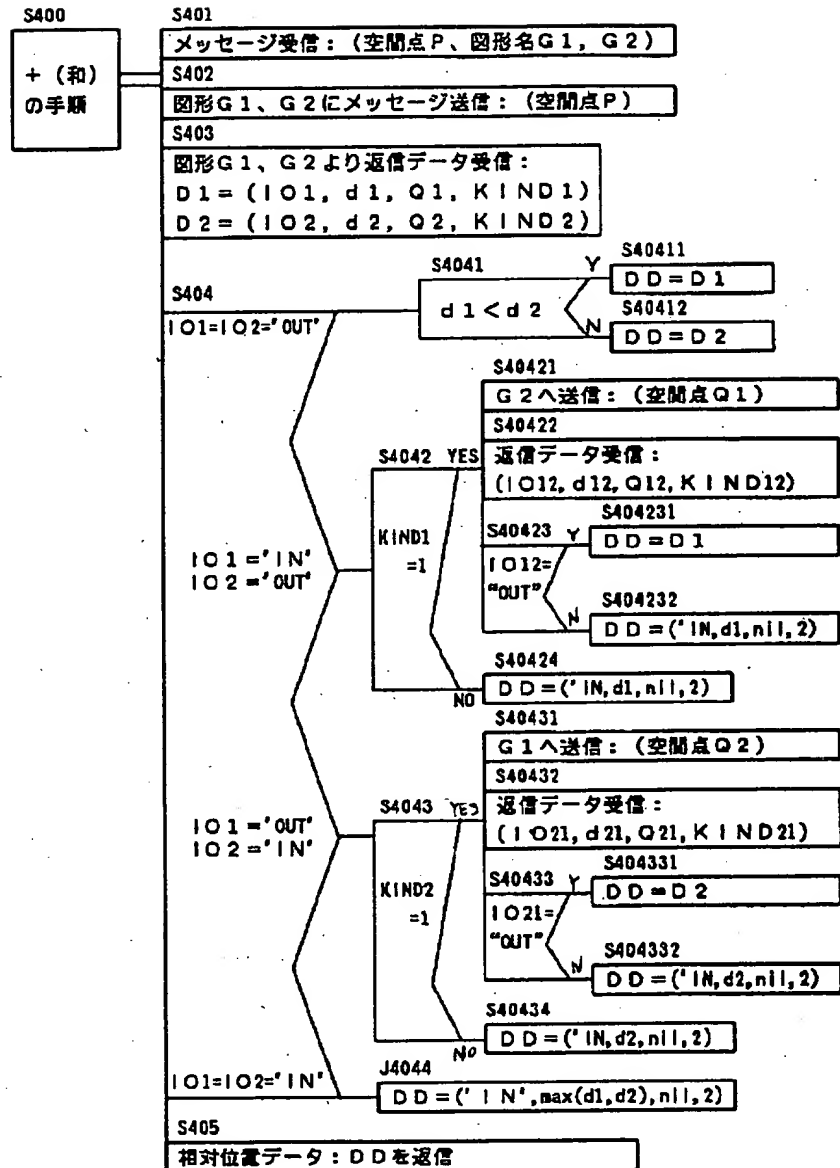
【図8】

図 8



【図11】

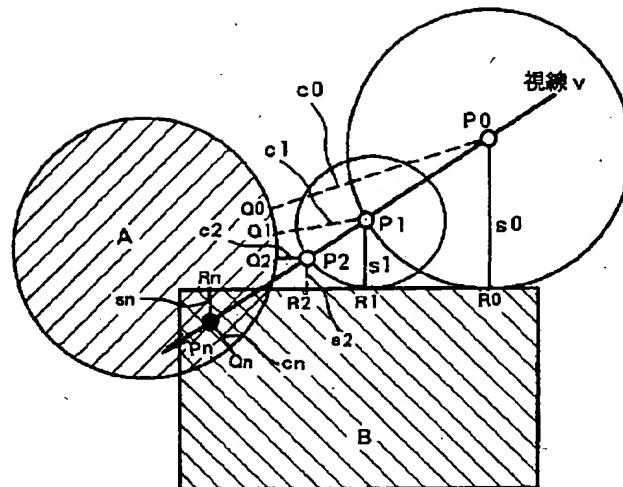
図 11



【図13】

図 13

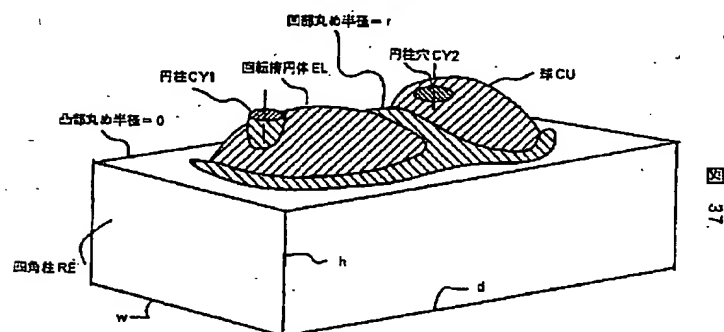
(a)



(b)

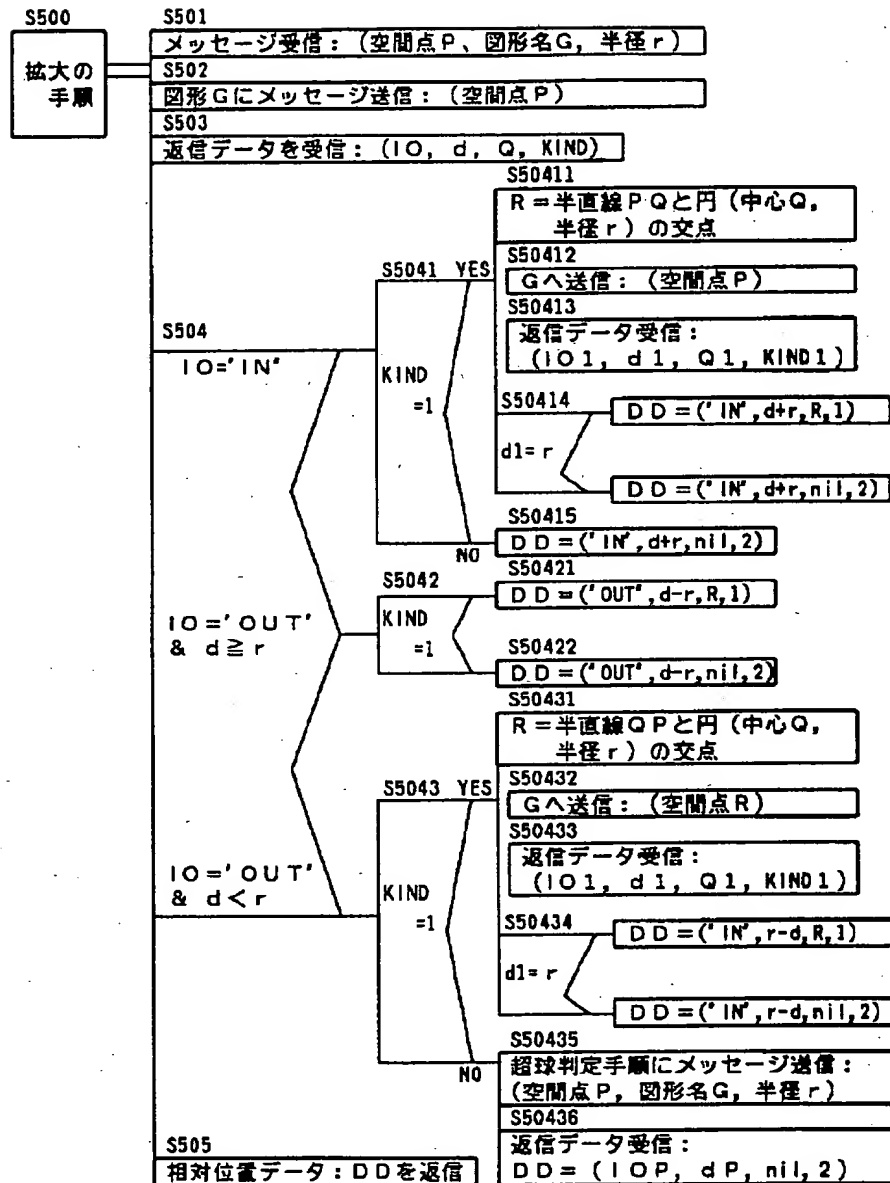
メッセージ 探索点 P	相対位置データ D1	相対位置データ D2	相対位置データ DD	備 考
P0 (開始点)	(OUT, c0, Q0, 1)	(OUT, s0, R0, 1)	(OUT, s0, R0, 1)	c0 > s0
P1(P0P1=c0)	(OUT, c1, Q1, 1)	(OUT, s1, R1, 1)	(OUT, s1, R1, 1)	c1 > s1
P2(P1P2=c1)	(OUT, c2, Q2, 1)	(OUT, s2, R2, 1)	(OUT, c2, Q2, 1)	s2 > c2
⋮	⋮	⋮	⋮	⋮
Pn	(OUT, cn, Qn, 1)	(IN, sn, Rn, 1)	(IN, cn, nil, 2)	cn > sn

【図37】



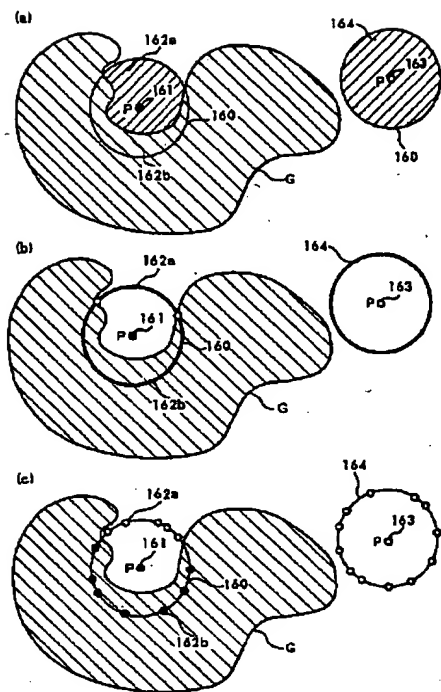
【図15】

図 15



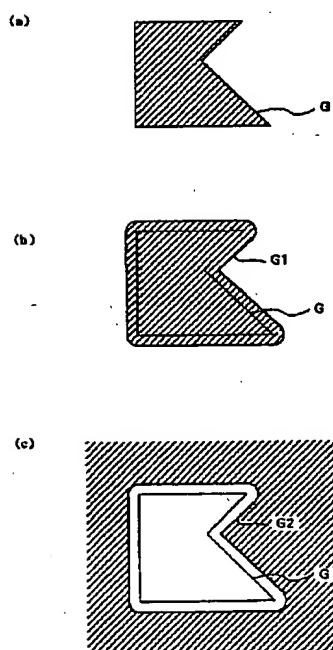
【図16】

図 16



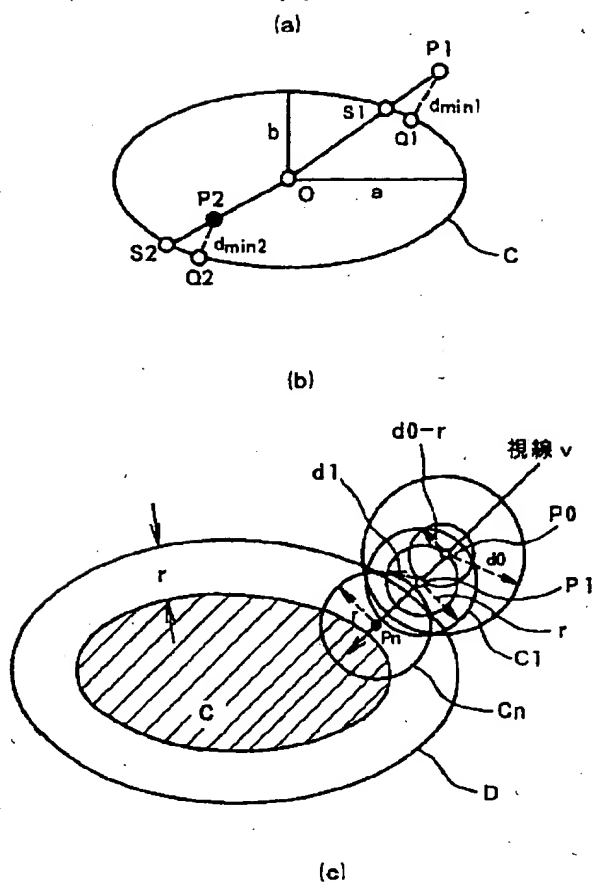
【図52】

図 52



【図19】

図 19

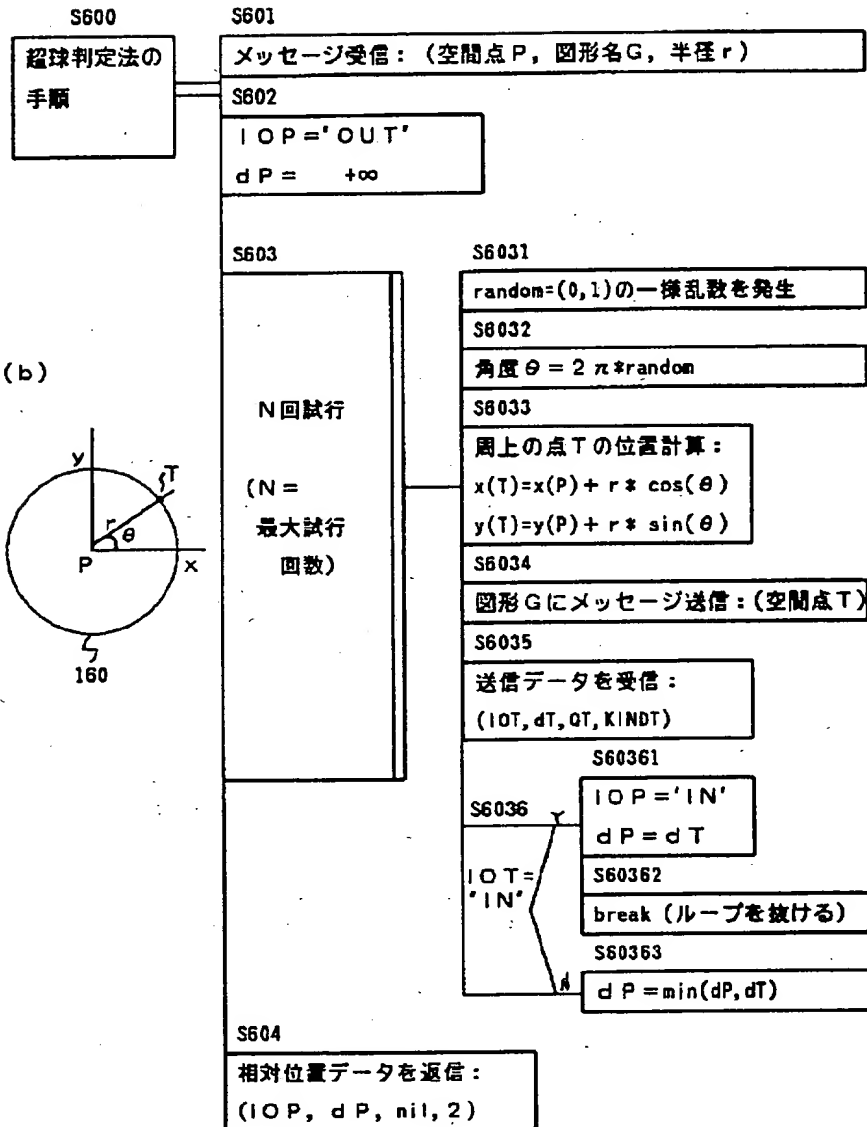


メッセージ 探索点 P	相対位置データ Dc	相対位置データ DD	備 考
P0(開始点)	(OUT, d0, nil, 2)	(OUT, d0-r, nil, 2)	$d0 > r$
P1(P0P1=d0-r)	(OUT, d1, nil, 2)	(OUT, s1, nil, 2)	$d1 < r$ により、円C1による 球判定法。s1はこれで決まる。
〜	〜	〜	
〜	〜	〜	
〜	〜	〜	
Pn	(OUT, dn, nil, 2)	(IN, sn, nil, 2)	$dn < r$ により、円Cnによる 球判定法。snはこれで決まる。

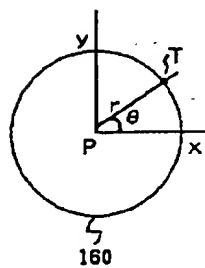
【図17】

図 17

(a)



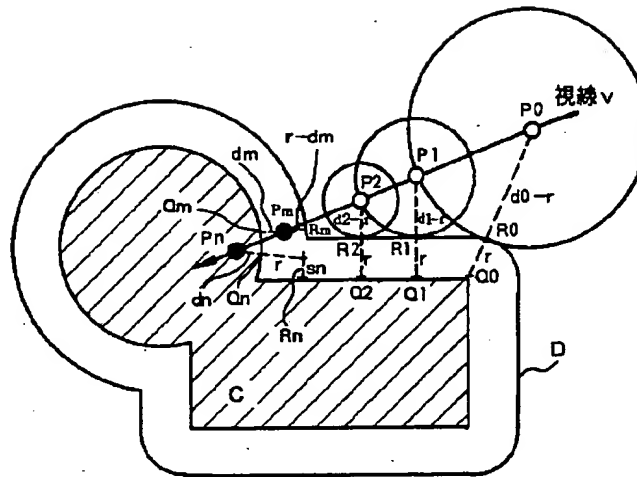
(b)



【図18】

図 18

(a)

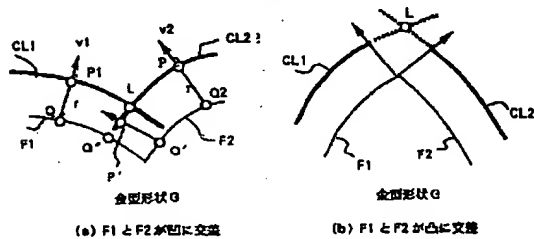


(b)

メッセージ 探索点 P	相対位置データ Dc	相対位置データ DD	備 考
P0(開始点)	(OUT, d0, Q0, 1)	(OUT, d0-r, R0, 1)	
P1(P0P1=d0-r)	(OUT, d1, Q1, 1)	(OUT, d1-r, R1, 1)	
P2(P1P2=d1-r)	(OUT, d2, Q2, 1)	(OUT, d2-r, R2, 1)	
~~~~~	~~~~~	~~~~~	
Pm	(OUT, dm, Qm, 1)	(IN, r-dm, Rm, 1)	Rmと図形Cの距離 = r
Pn	(IN, dn, Qn, 1)	(IN, dn+r, nil, 2)	Rnと図形Cの距離 < r

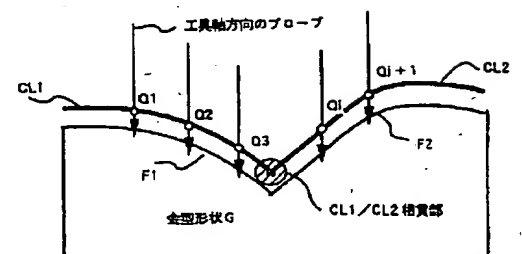
【図58】

図 58



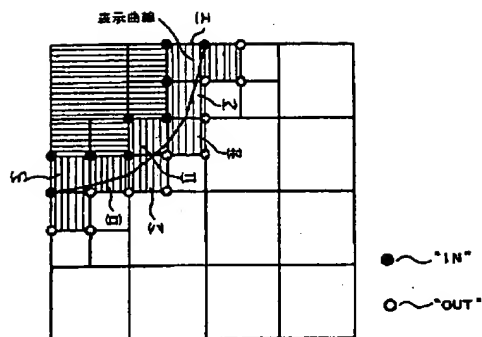
【図61】

図 61



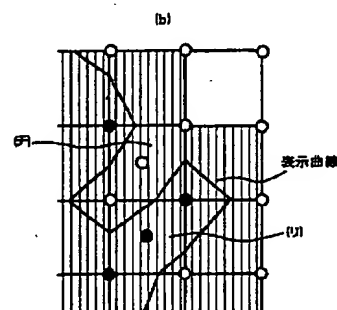
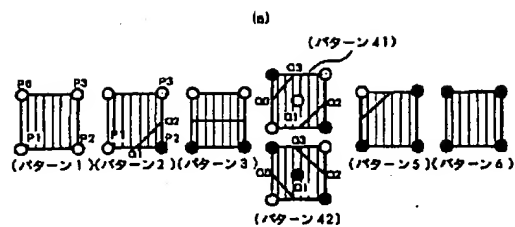
【図22】

図 22



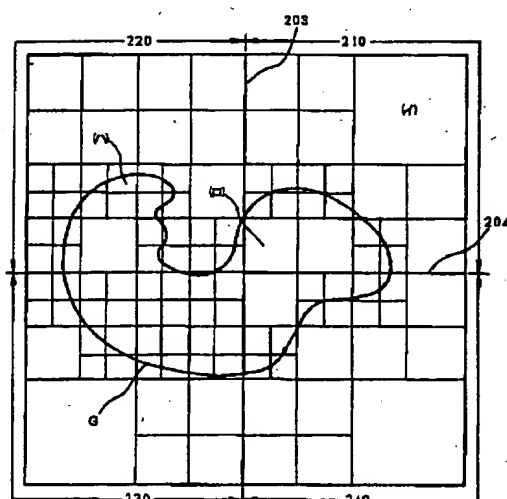
【図23】

図 23



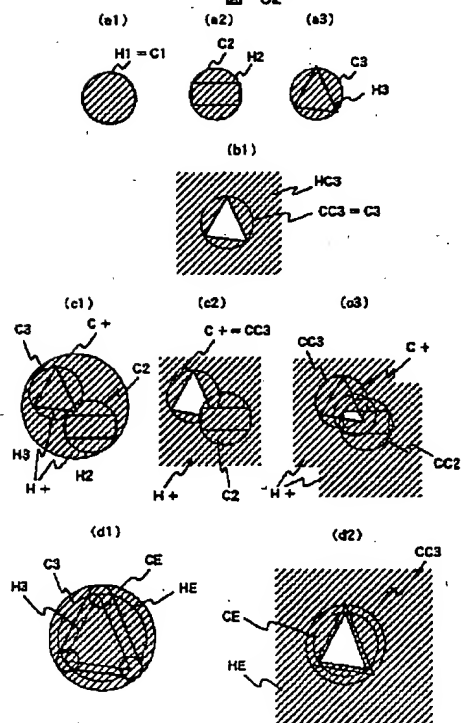
【図26】

図 26



【図32】

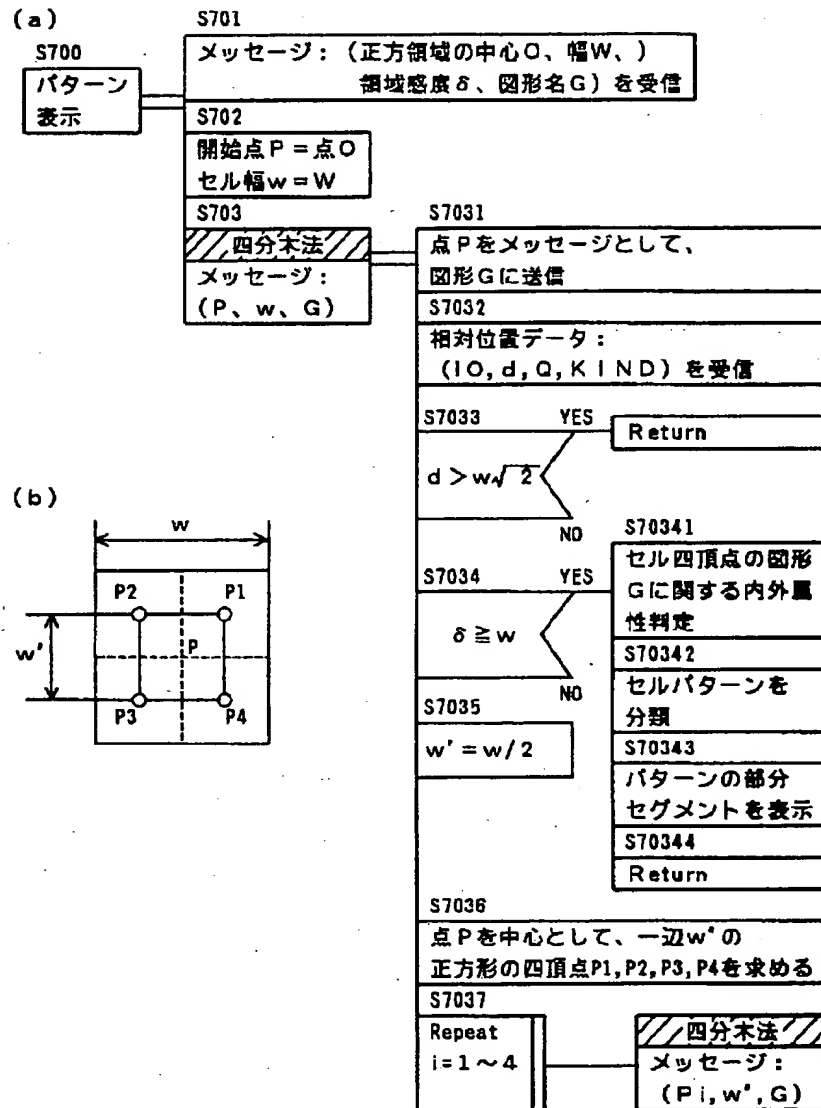
図 32





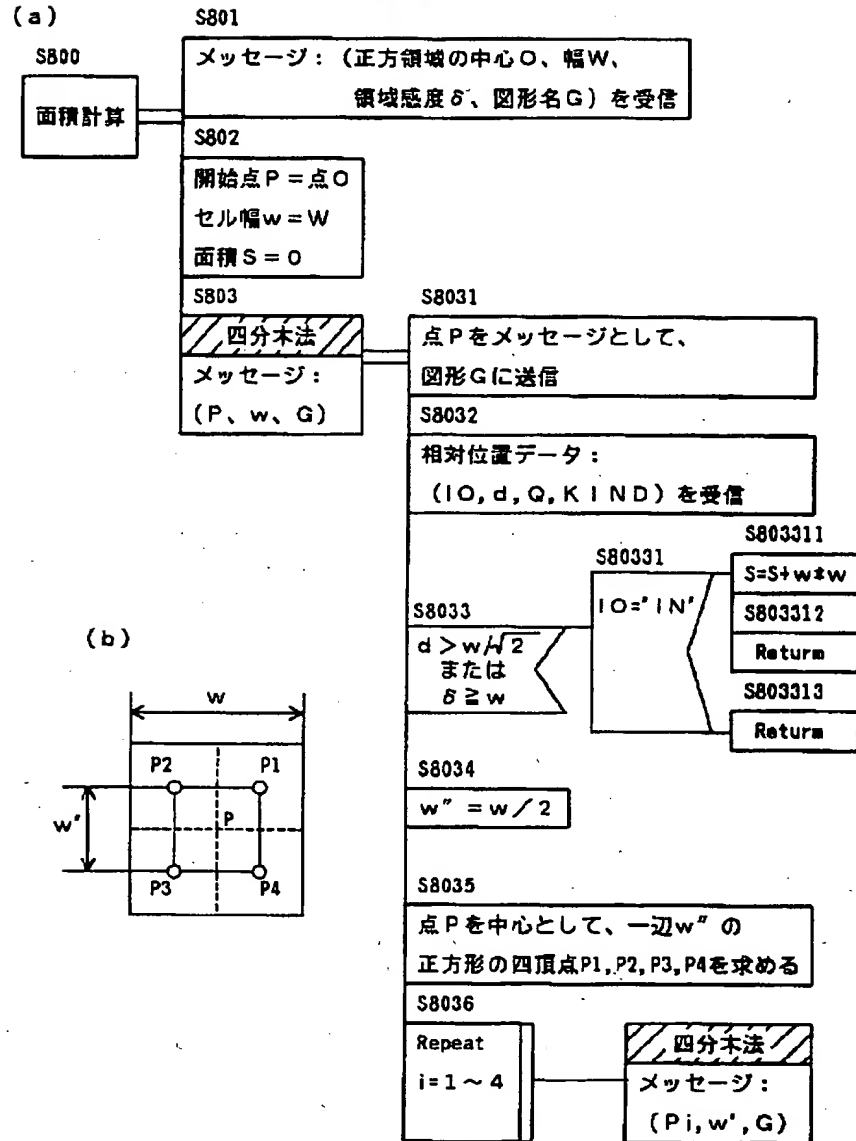
【図24】

図 24

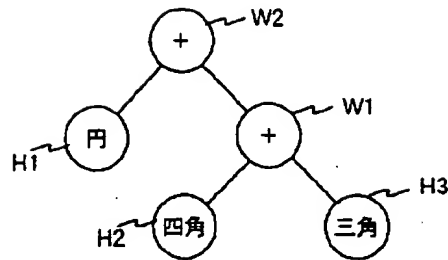


【図27】

図 27



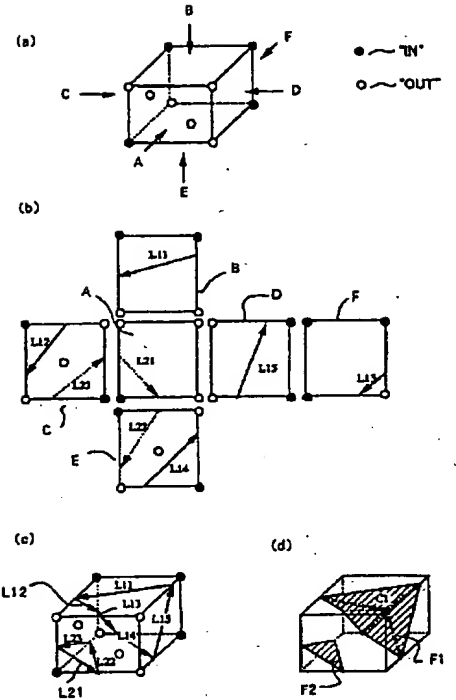
【図31】

図 31  
(a)(b)  
24

# obd2	図形名: W2	
	手順ポイント: # aaa	→ + (和) 図形生成手順
	図形ポイント: # obh1	
	図形ポイント: # obw1	
	領域データ	
# obh1	図形名: H1	
	手順ポイント: # ccc	→ 円生成手順
	パラメータ: 中心座標値と半径の値	
	領域データ	
# obd1	図形名: W1	
	手順ポイント: # aaa	→ + (和) 図形生成手順
	図形ポイント: # obh2	
	図形ポイント: # obh3	
	領域データ	
# obh2	図形名: H2	
	手順ポイント: # sss	→ 四角形生成手順
	パラメータ: 中心座標値と二辺の長さ	
	領域データ	
# obh3	図形名: H3	
	手順ポイント: # ttt	→ 三角形生成手順
	パラメータ: 3 頂点の座標値	
	領域データ	

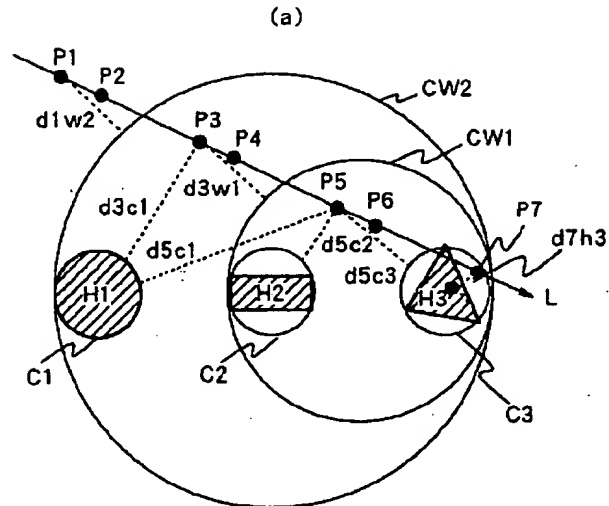
【図48】

図 48



【図33】

図 33



dihj...点Piと図形Hjとの距離      diwj...点Piと円CWjとの距離  
 dicj...点Piと円Cjとの距離

(b)

図形名	W2 (+)	H1	W1 (+)	H2	H3
領域データ	(OUT, 円CW2)	(OUT, 円C1)	(OUT, 円CW1)	(OUT, 円C2)	(OUT, 円C3)
相対位置データ	P1 (OUT, d1w2, nil, 2)				
	P2 (OUT, d2w2, nil, 2)				
	P3 (OUT, d3w1, nil, 2)	(OUT, d3c1, nil, 2)	(OUT, d3w1, nil, 2)		
	P4 (OUT, d4w1, nil, 2)	(OUT, d4c1, nil, 2)	(OUT, d4w1, nil, 2)		
	P5 (OUT, d5c2, nil, 2)	(OUT, d5c1, nil, 2)	(OUT, d5c2, nil, 2)	(OUT, d5c2, nil, 2)	(OUT, d5c3, nil, 2)
	P6 (OUT, d6c3, nil, 2)	(OUT, d6c1, nil, 2)	(OUT, d6c3, nil, 2)	(OUT, d6c2, nil, 2)	(OUT, d6c3, nil, 2)
	P7 (OUT, d7h3, Q, 1)	(OUT, d7c1, nil, 2)	(OUT, d7h3, Q, 1)	(OUT, d7c2, nil, 2)	(OUT, d7h3, Q, 1)

計算なし

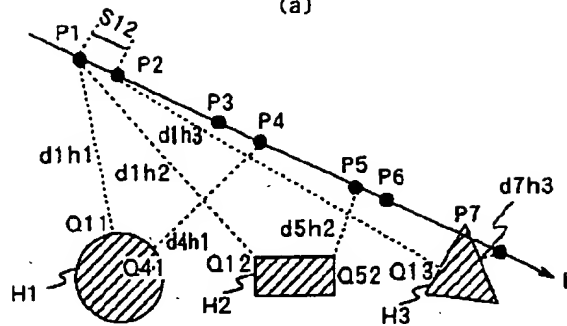
xxxx 簡略計算

xxxx 詳細計算

【図34】

図 34

(a)



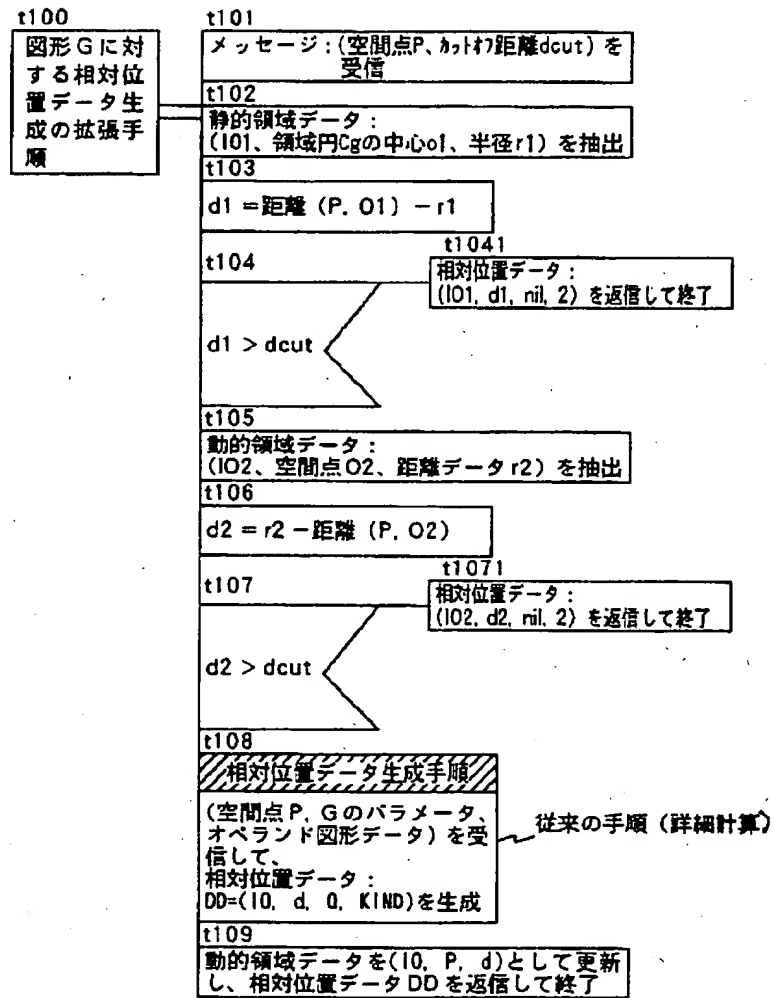
(b)

図形名	W2 (+)	H1	W1 (+)	H2	H3
上段：相対位置データ 下段：領域データ	P1	(OUT, d1h1, Q11, 1) (OUT, P1, d1h1)	(OUT, d1h2, Q12, 1) (OUT, P1, d1h2)	(OUT, d1h3, Q13, 1) (OUT, P1, d1h3)	
	P2	(OUT, d1h1 - s12, nil, 2)			
	P3	(OUT, d1h1 - s13, nil, 2)			
	P4	(OUT, d1h2 - s14, nil, 2) (OUT, P4, d1h2 - s14)	(OUT, d4h1, Q41, 1) (OUT, P4, d4h1)		
	P5	(OUT, d4h1 - s45, nil, 2) (OUT, P5, d4h1 - s45)	(OUT, d5h2, Q52, 1) (OUT, P5, d5h2)	(OUT, d5h2, Q52, 1) (OUT, P5, d5h2)	(OUT, d1h3 - s15, nil, 2)
	P6	(OUT, d4h1 - s46, nil, 2) (OUT, P6, d4h1 - s46)	(OUT, d5h2 - s56, nil, 2) (OUT, P6, d5h2 - s56)	(OUT, d5h2 - s56, nil, 2) (OUT, P6, d5h2 - s56)	(OUT, d1h3 - s16, nil, 2)
	P7	(OUT, d7h3, Q73, 1) (OUT, P7, d7h3)	(OUT, d7h3, Q73, 1) (OUT, P7, d7h3)	(OUT, d7h2, Q72, 1) (OUT, P7, d7h2)	(OUT, d7h3, Q73, 1) (OUT, P7, d7h3)

(相対位置データ) 計算なし	xxxxxxx	簡略計算	xxxxxxx	詳細計算
(領域データ) 更新なし		更新なし	xxxxxxx	更新あり

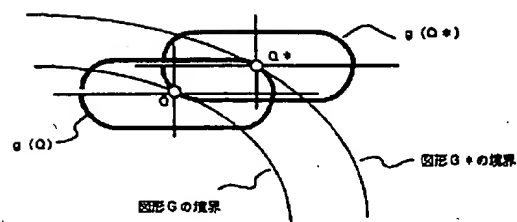
【図35】

図 35



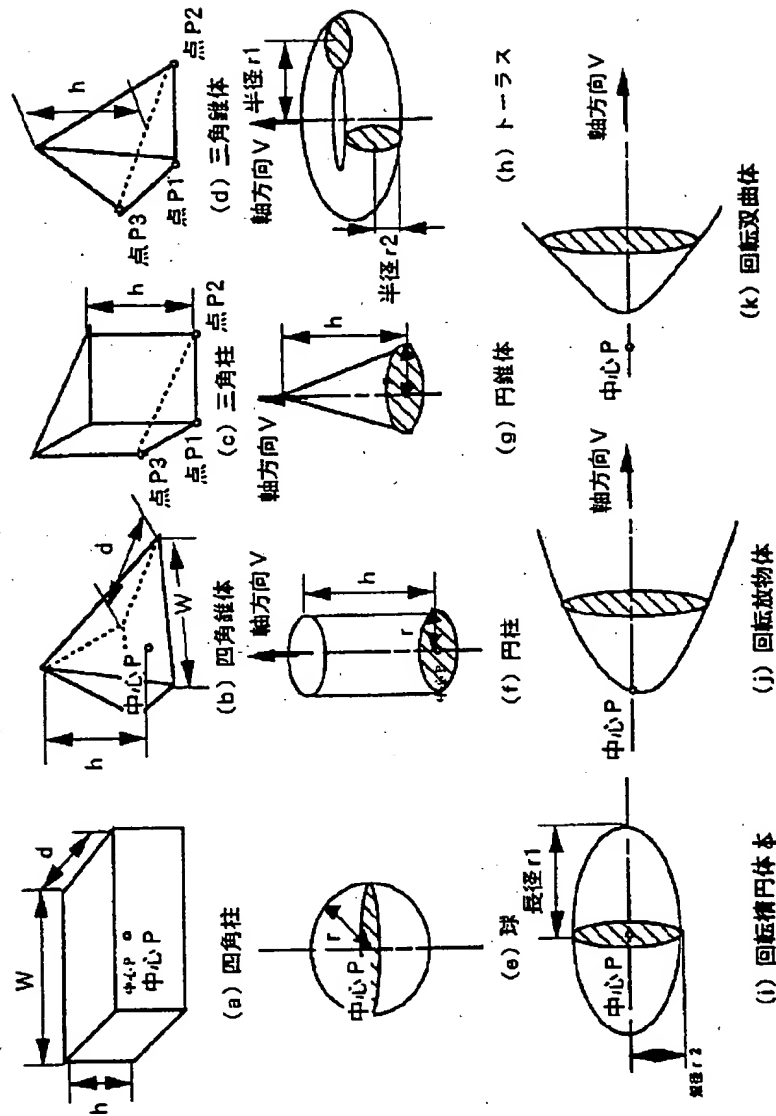
【図62】

図 62



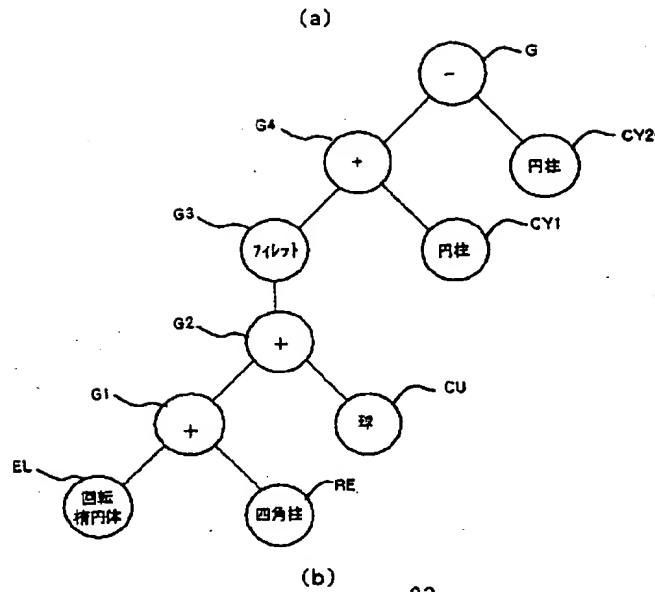
【図36】

図 36



【図38】

図 38

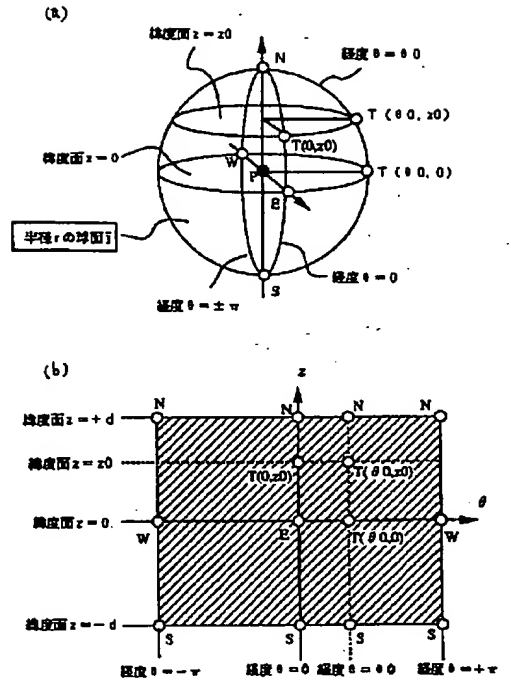


23

# fff	球の相対位置生成手順
# ggg	円柱の "
# hhh	四角柱の "
# iii	回転楕円体の "
	}
# jjj	和(+)図形の相対位置生成手順
# kkk	差(-)図形の "
# lll	フィレット図形の "
	}

【図41】

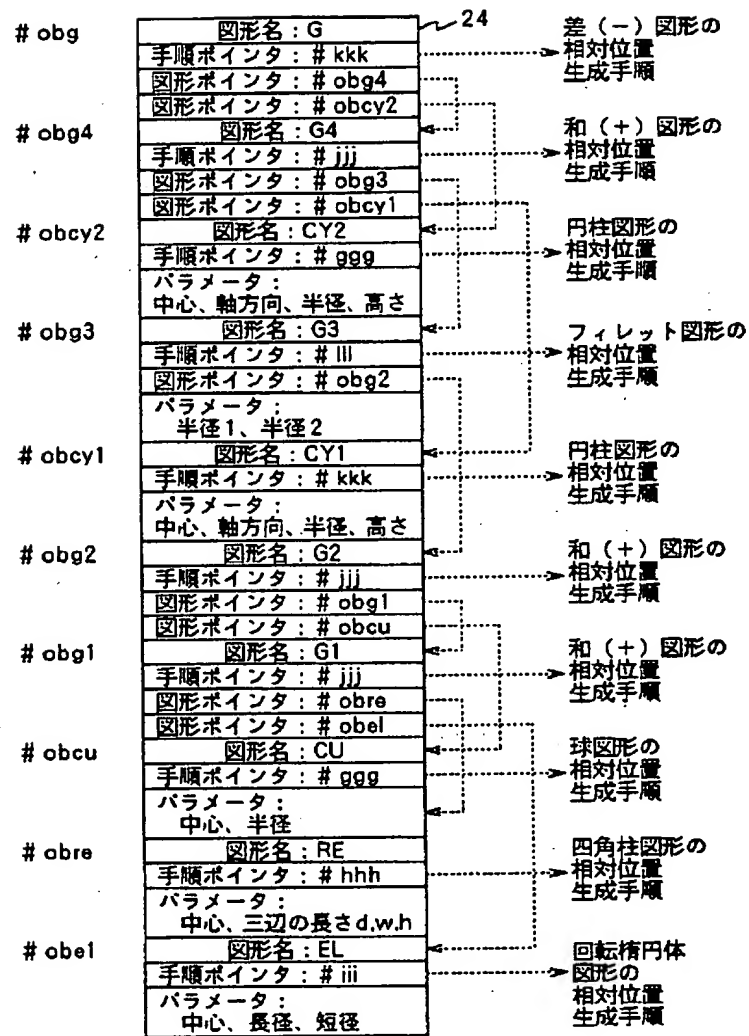
図 41





【図39】

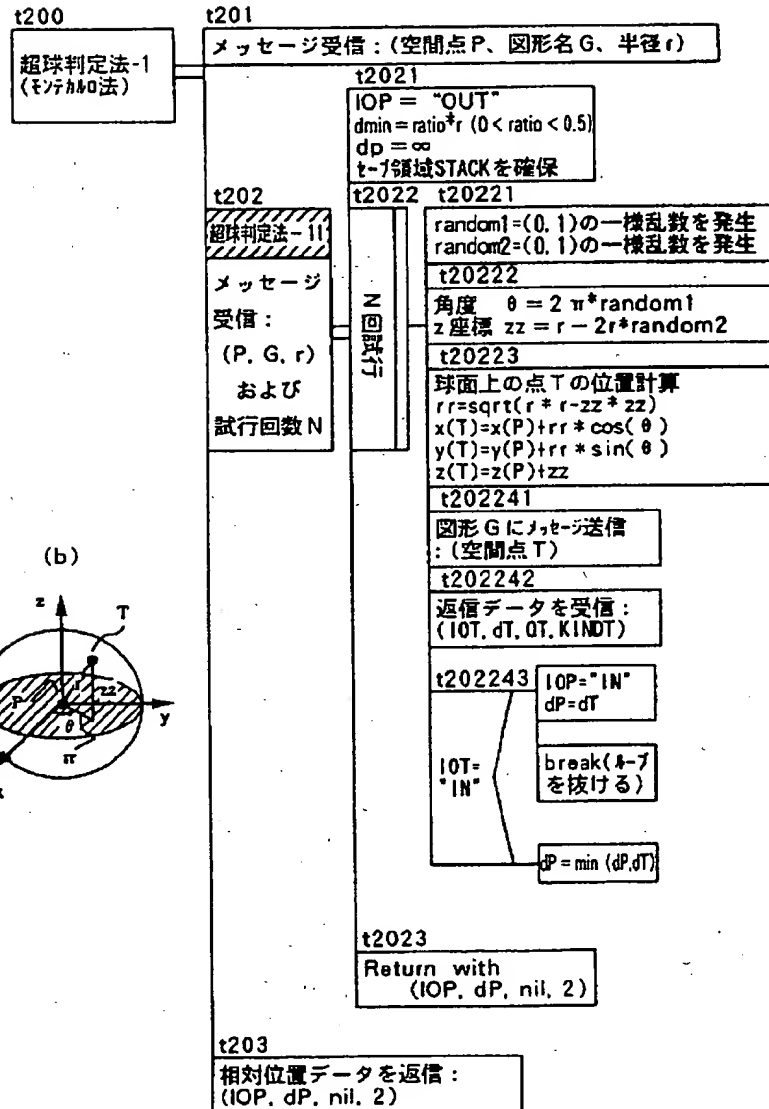
図 39



【図40】

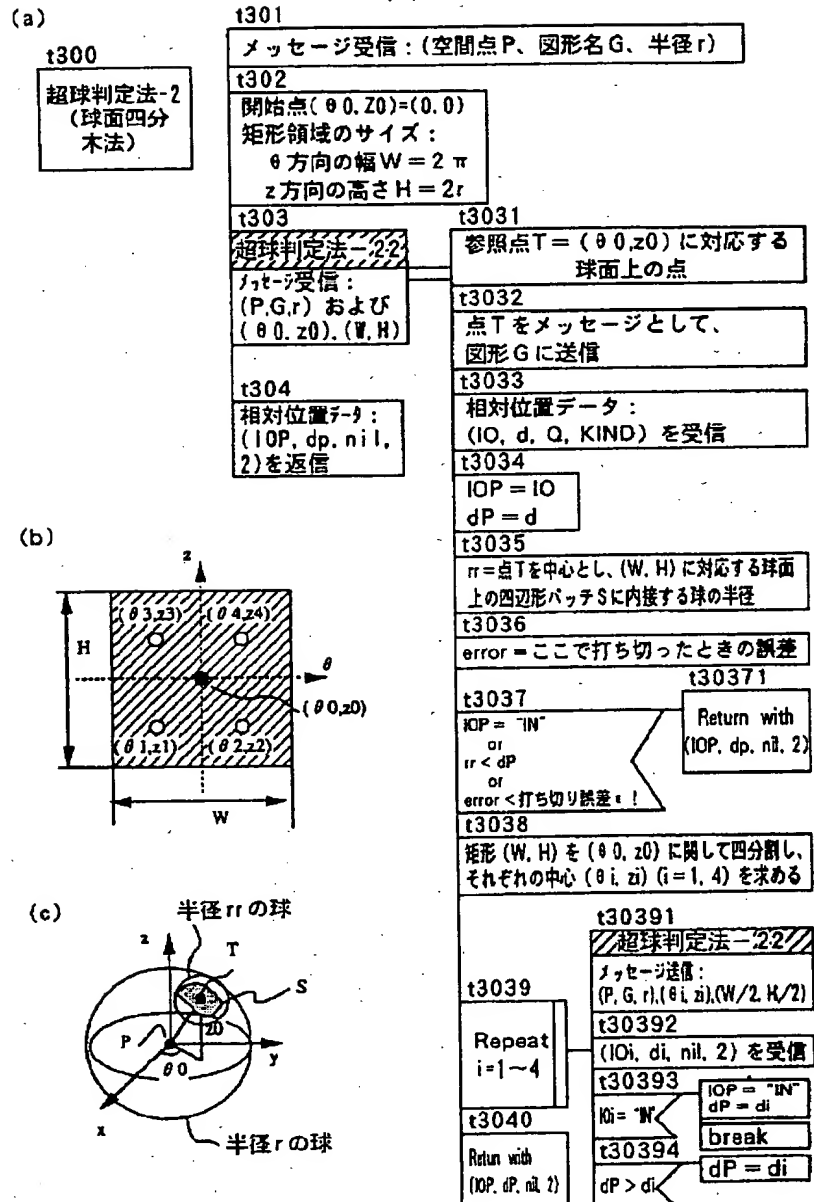
図 40

(a)



【図42】

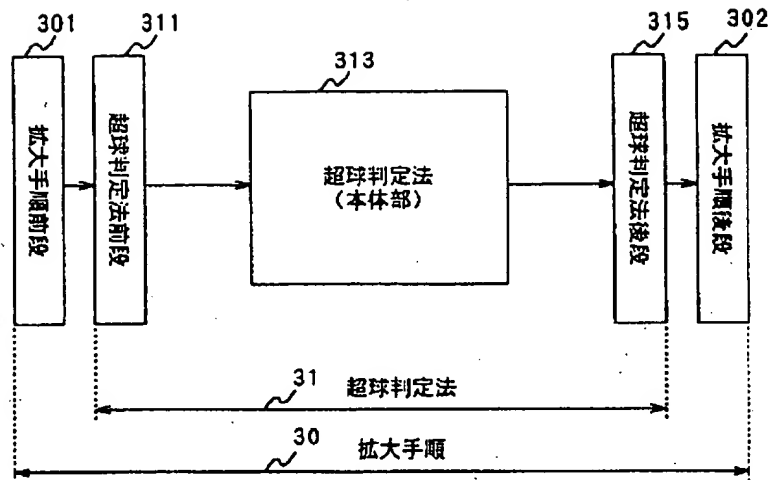
図 42



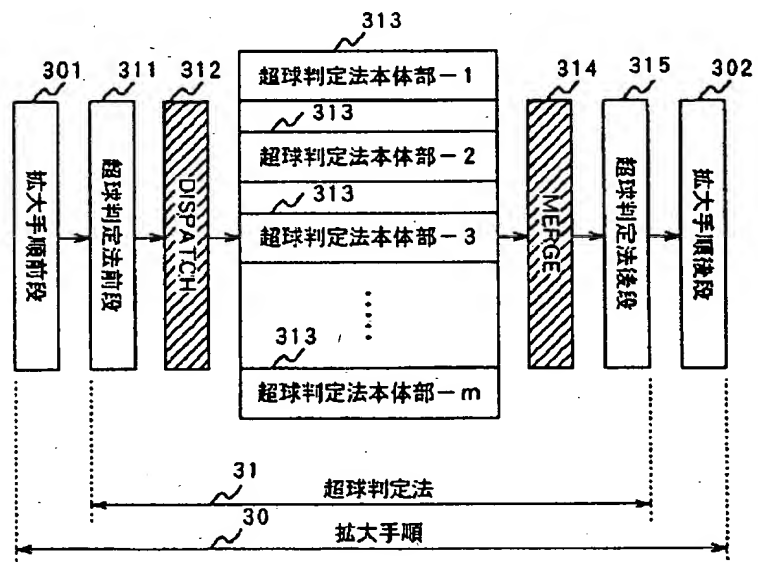
【図43】

図 43

(a)

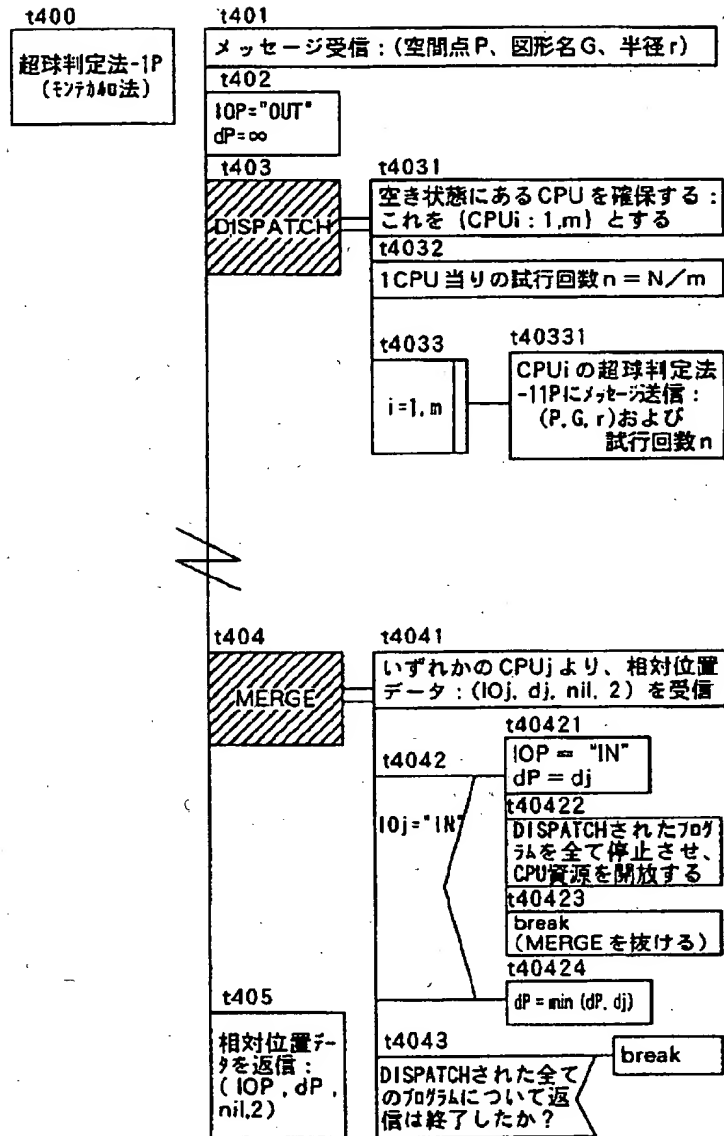


(b)



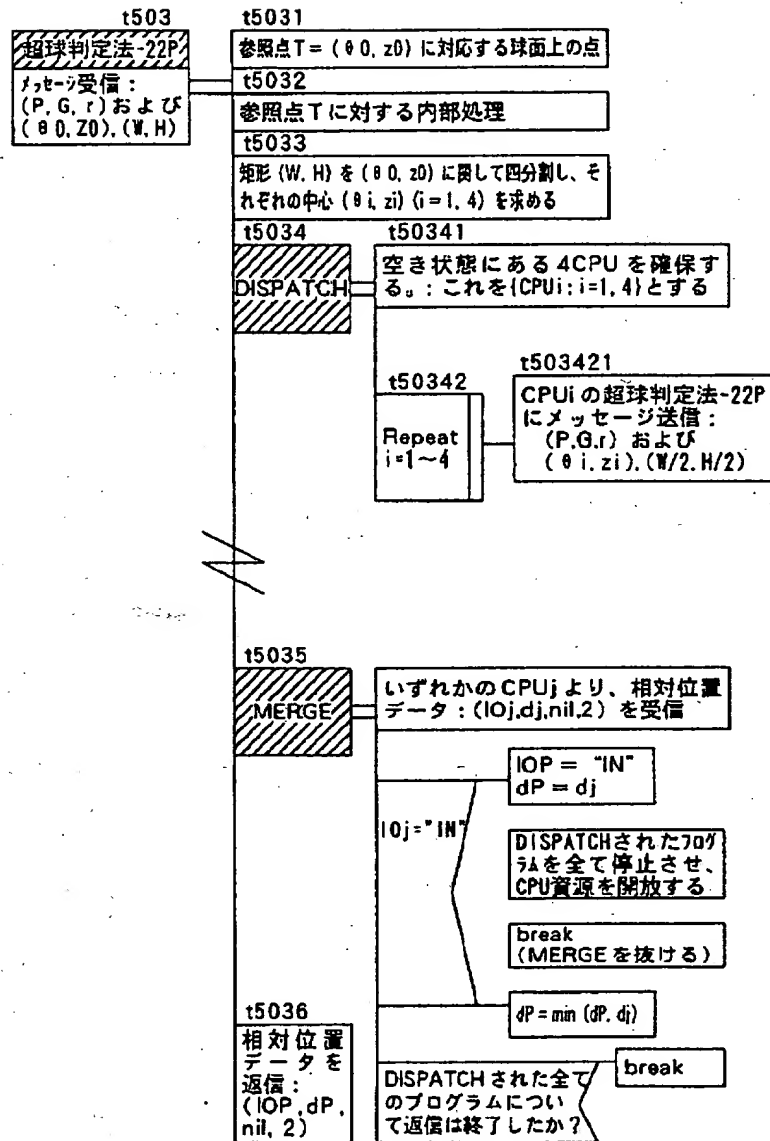
【図44】

図 44



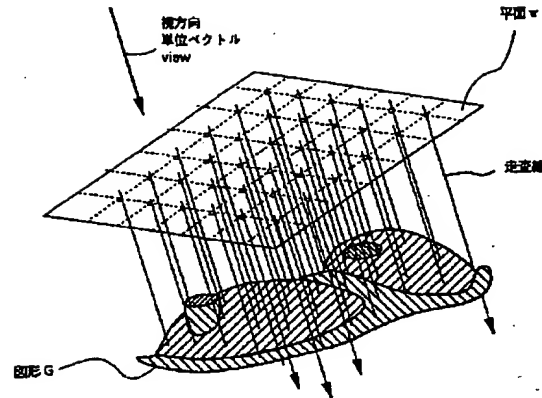
【図45】

図 45



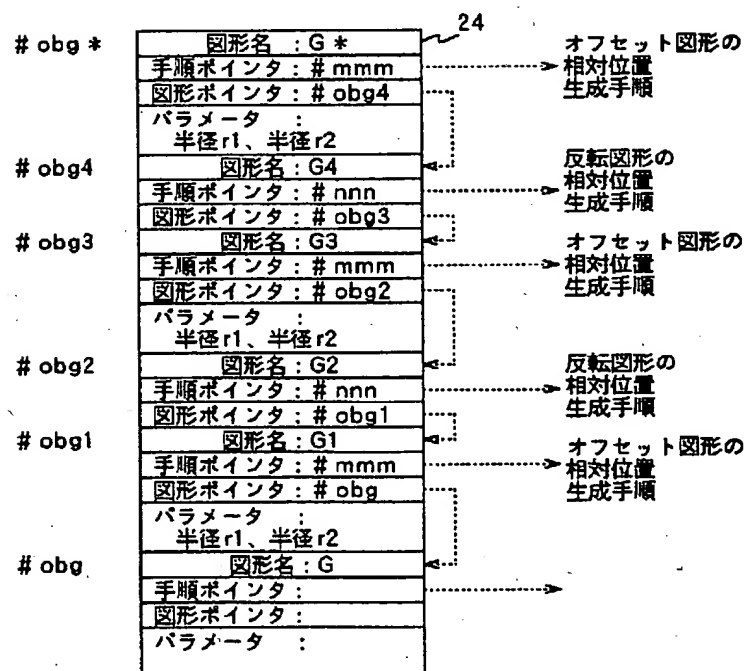
【図46】

図 46



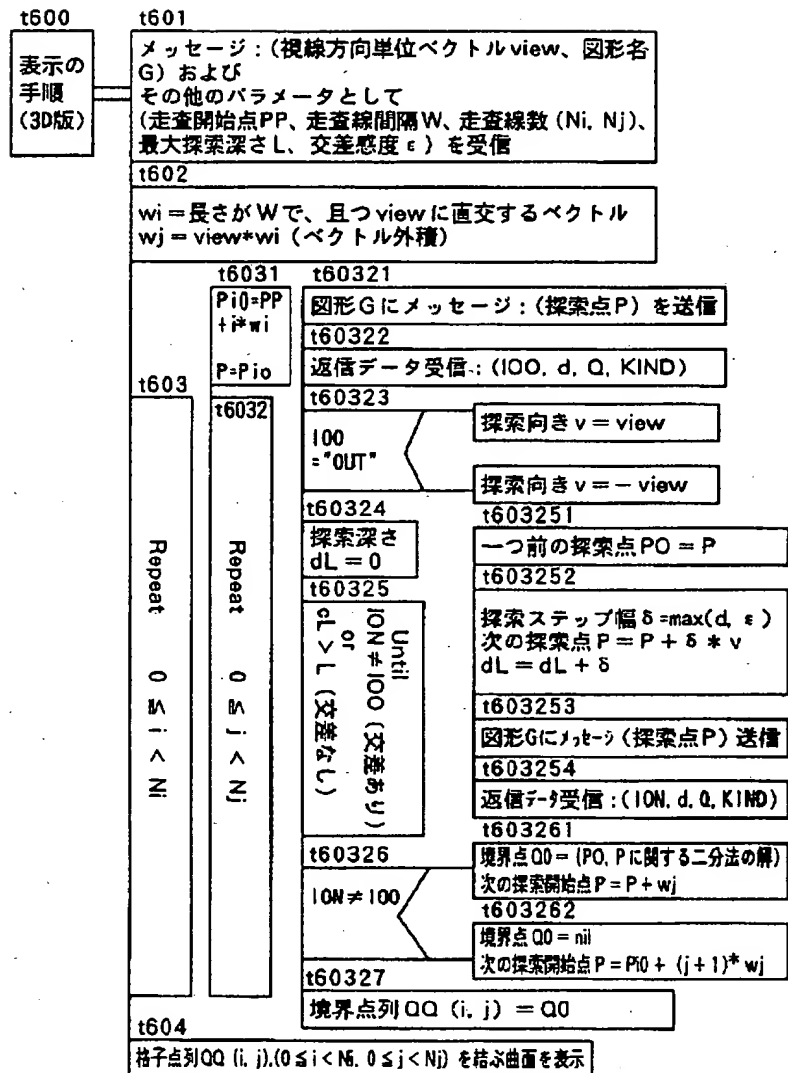
【図55】

図 55



【図47】

図 47





【図50】

図 50

(a)

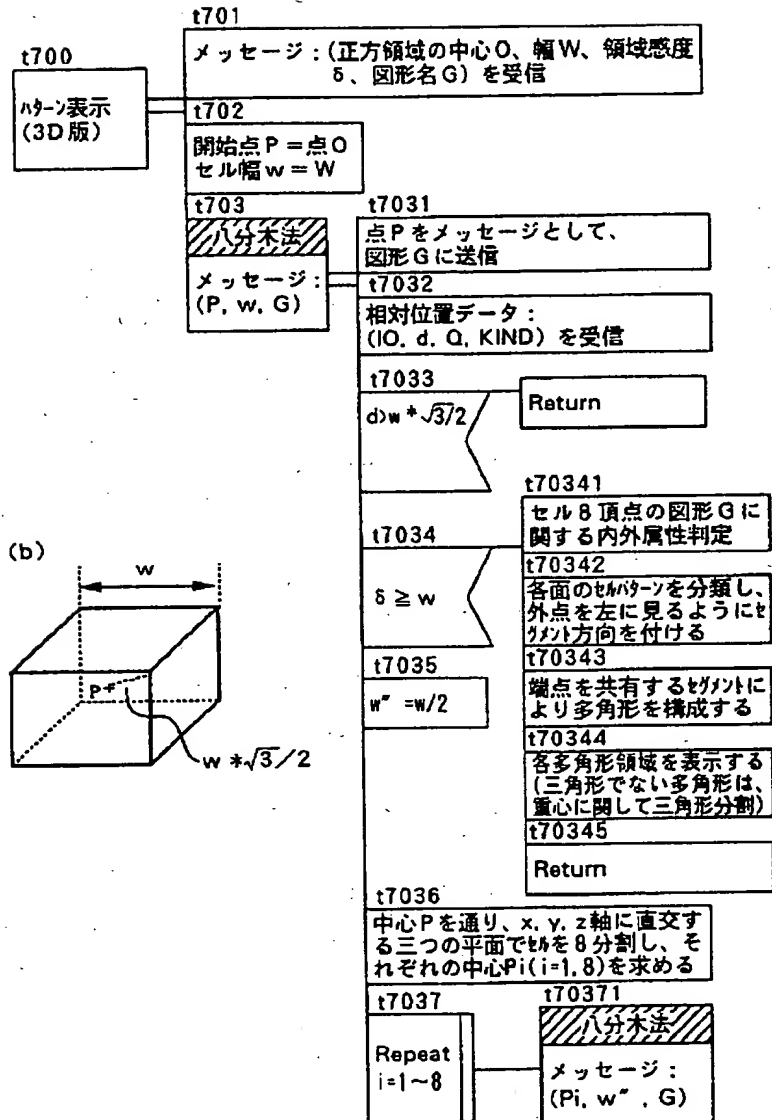


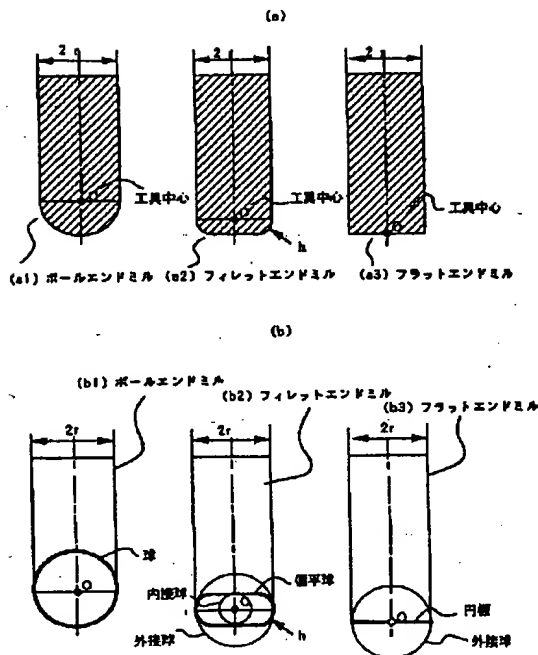
图 51

(b)

Diagram (b) shows a 3D rectangular prism. The width of the prism is labeled  $w$ . The height of the prism is labeled  $w \cdot \sqrt{3}/2$ . A point  $P$  is marked on the front face of the prism. Dashed lines indicate the dimensions and the point's position.

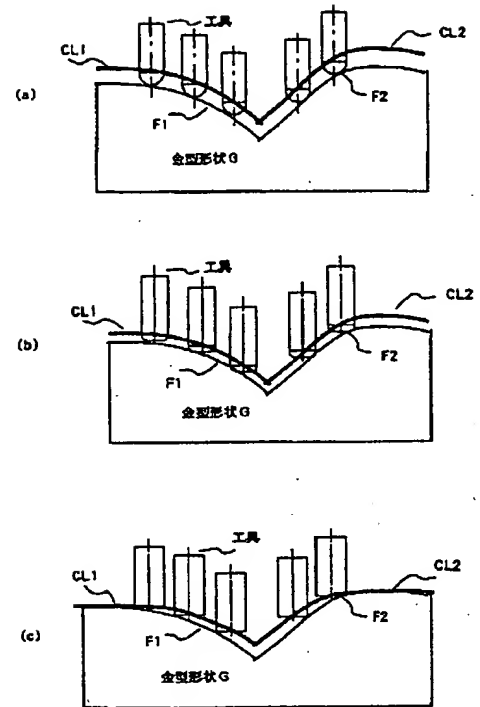
【図56】

図 56



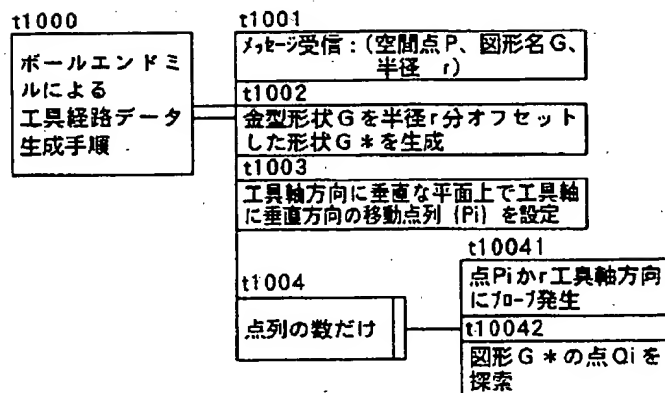
【図59】

図 59



【図60】

図 60



【図63】

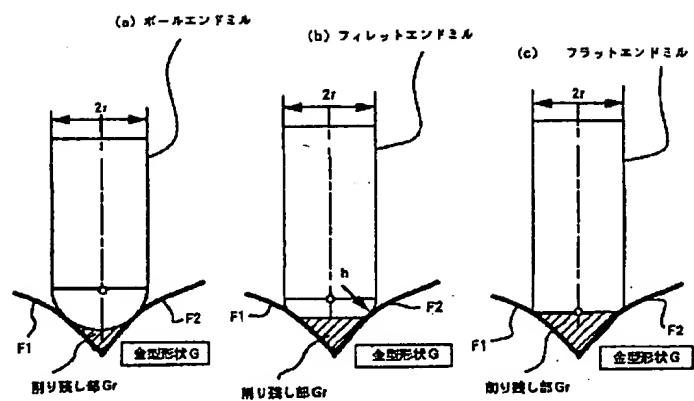


図 63